



# Couplage mécano-fiabiliste : 2 SMART - méthodologie d'apprentissage stochastique en fiabilité

François Deheeger

## ► To cite this version:

François Deheeger. Couplage mécano-fiabiliste : 2 SMART - méthodologie d'apprentissage stochastique en fiabilité. Génie mécanique [physics.class-ph]. Université Blaise Pascal - Clermont-Ferrand II, 2008. Français. NNT : 2008CLF21822 . tel-00728322

**HAL Id: tel-00728322**

**<https://theses.hal.science/tel-00728322>**

Submitted on 5 Sep 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : D.U. : 1822

EDSPIC : 398

Université BLAISE PASCAL - Clermont II

École Doctorale

Sciences pour l'Ingénieur de Clermont-Ferrand

**T H È S E**

présentée et soutenue publiquement par

**François DEHEEGER**

Ingénieur IFMA

le **28 janvier 2008**

pour obtenir le grade de

**Docteur d'Université**

(Spécialité : Génie Mécanique )

**Couplage mécano-fiabiliste :**

**<sup>2</sup>SMART - méthodologie d'apprentissage**

**stochastique en fiabilité**

Jury :

ALEXIS BEAKOU	IFMA, Clermont-Ferrand	Président du jury
JEAN-MARC BOURINET	IFMA, Clermont-Ferrand	Encadrant
FABIEN FESCHET	Université d'Auvergne de Clermont-Ferrand	Rapporteur
LAURENT GUILLAUMAT	ENSAM de Bordeaux	Rapporteur
MAURICE LEMAIRE	IFMA, Clermont-Ferrand	Directeur de thèse
PIERRE VILLON	Université Technologique de Compiègne	Rapporteur

**Laboratoire de Mécanique et Ingénieries (LaMI)**

**Université Blaise Pascal et Institut Français de Mécanique Avancée**







*À Lou,  
À mes familles...  
Celle qui m'a vu grandir...  
Celle que je vois grandir.*



# Remerciements

Il est d'usage de commencer par remercier la direction de la thèse. Je ne dérogerais pas à cette règle mais en plus de saluer le guide stratégique-scientifique, je voulais insister sur les aspects humains : j'ai en effet réellement apprécié les 3 années passées sous la tutelle du professeur Maurice Lemaire... depuis les discussions sur les outils de l'apprentissage statistique au partage de plateaux repas sur la baie de Tokyo. Je lui exprime ma sincère gratitude pour la confiance exprimée pendant la thèse.

Je remercie chaleureusement les collègues de travail : collègues du LaMI, les voisins plus ou moins proches de l'open-space du bout du couloir ; ou d'ailleurs : collègues de Phimeca et d'EdF, du LAIC, pour les échanges - de tous ordres - qui m'ont permis d'avancer.

Je pense également à ma famille, parents, frères et soeur... Ainsi qu'à mes très proches, Martine qui a accepté de vivre 3 années de plus avec un étudiant et Lou qui a attendrit les membres du jury à la seconde où j'attaquais l'introduction... 'Papa ?'

Merci à tous ceux qui m'ont soutenu.

Bonne lecture à ceux qui commencent.





# Résumé

La prise en compte de l'incertain dans les analyses mécaniques est une condition indispensable pour un dimensionnement optimal et robuste des structures. C'est avec cet objectif que les approches probabilistes pour la mécanique des structures sont développées depuis plusieurs décennies. Ces méthodes permettent en effet d'étudier la fiabilité de composants ou de systèmes et l'influence de la variabilité des variables de conception sur leur comportement.

Les méthodes de couplage mécano-fiabiliste se doivent alors d'intégrer les modélisations mécaniques de plus en plus complexes (comportement non-linéaire, dynamique, fatigue, mécanique de la rupture, etc), pour rendre efficace et exploitable les études de fiabilité. Cette nécessaire complexité rend les analyses de plus en plus exigeantes en temps de calcul.

La voie retenue dans la thèse consiste à substituer au modèle mécanique, un modèle approché obtenu par apprentissage statistique. L'efficacité est alors mesurée par le nombre des calculs mécaniques nécessaires pour construire un substitut robuste pour l'analyse visée : fiabilité ou sensibilité.

La mise en œuvre s'appuie sur la méthode des *Support Vector Machines* (SVM) dont l'application en classification est particulièrement bien adaptée au calcul de la probabilité de défaillance, même lorsque la fonction d'état-limite présente des singularités. Deux nouvelles méthodes sont proposées. *SMART Support vector Margin Algorithm for Reliability esTimation* permet d'élaborer de manière incrémentale une base de données optimale pour l'apprentissage de la frontière entre sûreté et défaillance. En couplant l'apprentissage avec la technique des *subsets simulations*, <sup>2</sup>*SMART Subset by Support vector Margin Algorithm for Reliability esTimation* prend en compte de faibles niveaux de probabilité et un nombre relativement important de variables.

Les travaux ont été validés sur des cas tests issus de la littérature ainsi que sur des cas mécaniques plus significatifs issus de problèmes industriels. Les résultats obtenus confirment la robustesse des méthodes proposées pour traiter les analyses de fiabilité et de sensibilité.



# Abstract

Uncertainties usually play an important role in structural engineering and their effects need to be assessed in a rational way in order to meet performance, robustness and safety requirements. Accounting for uncertainties is usually carried out within a probabilistic framework and corresponding approaches fall under the so called fields of structural reliability and stochastic structural mechanics.

Coupling mechanical and stochastic data models for complex systems using a classical procedure may become prohibitive in terms of computer time. This time increases as the probability of failure of interest decreases, and this is a major limitation in using these methods for industrial applications. This is especially true of simulation methods which are unfortunately the reference standard in terms of accuracy. S. Au and J. Beck address the problem of small probabilities by introducing the subset simulation method, but such a method is still a simulation. To overcome the computing time difficulty, limit-state substitution is proposed.

J.E. Hurtado (Structural Reliability - Statistical Learning Perspectives) introduces Support Vector Machine to reliability and applies their classification properties to produce an efficient Monte-Carlo simulation. In SMART, we improved his algorithm to build the limit-state surrogate for reliability application. By merging Support Vector Machine classification into subset simulation, we propose here a robust method for reliability analysis referred as <sup>2</sup>SMART for Subset by Support vector Margin Algorithm for Reliability esTimation.

The proposed methodology is detailed and its efficiency is illustrated through various examples, from test cases for validation to structural mechanics applications.



# Table des matières

<b>Remerciements</b>	<b>iv</b>
<b>Résumé</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Table des matières</b>	<b>xiii</b>
<b>Introduction</b>	<b>1</b>
<b>I Méthodes de calcul de la fiabilité des structures</b>	<b>5</b>
I.1 Introduction . . . . .	7
I.2 Un problème de fiabilité . . . . .	8
I.2.1 La fiabilité en mécanique. . . . .	8
I.2.2 . . .une coopération entre divers acteurs . . . . .	8
I.3 Le calcul mécano-fiabiliste . . . . .	10
I.3.1 La fonction d'état-limite . . . . .	10
I.3.2 La mesure d'une probabilité . . . . .	10
I.3.3 Deux familles de méthodes . . . . .	11
I.3.4 Substitution de l'état-limite . . . . .	12
I.4 Transformation des variables de base . . . . .	12
I.5 FORM et SORM . . . . .	13
I.5.1 Le concept . . . . .	13
I.5.2 Approximation de Taylor . . . . .	13
I.5.3 La recherche du point de conception . . . . .	14
I.5.4 Le calcul des dérivées . . . . .	15
I.5.5 Les limites du calcul FORM/SORM . . . . .	16
I.6 Améliorer le calcul des gradients . . . . .	16

I.6.1	Méthode de différentiation directe . . . . .	17
I.6.2	Implémentation de la DDM . . . . .	18
I.6.3	Approximant de Padé [Wik07d] . . . . .	21
I.7	Les méthodes de simulation . . . . .	22
I.7.1	<i>Crude</i> Monte-Carlo . . . . .	22
I.7.2	Simulations d'importance . . . . .	24
I.7.3	Simulations directionnelles . . . . .	25
I.7.4	Carré Latin . . . . .	25
I.7.5	Subset Simulations . . . . .	26
I.7.6	Conclusion . . . . .	27
I.8	Substituer la fonction de performance . . . . .	27
I.8.1	Pourquoi substituer l'état-limite ? . . . . .	27
I.8.2	Surfaces de réponse polynomiales . . . . .	28
I.8.3	Réseaux de neurones ( <i>Neural Network</i> NN) . . . . .	31
I.8.4	Classification et régression par les SVM . . . . .	33
I.9	Bilan . . . . .	34
<b>II</b>	<b>Machines à Vecteurs Supports</b>	<b>37</b>
II.1	Introduction . . . . .	39
II.2	Classification et SVM . . . . .	39
II.2.1	Le cas linéaire . . . . .	40
II.2.2	Le cas non linéaire - méthode à noyaux . . . . .	43
II.2.3	Les classes non séparables . . . . .	45
II.3	Régression par SVM . . . . .	47
II.4	SVM ou réseaux de neurones . . . . .	49
II.4.1	ERM et principe de régularisation . . . . .	50
II.4.2	VC dimension et SRM . . . . .	54
II.4.3	La grande dimension . . . . .	57
II.5	Optimisation des SVM . . . . .	58
II.5.1	Le choix des paramètres . . . . .	58
II.5.2	Validation croisée . . . . .	59
II.5.3	<i>Boosting</i> algorithm . . . . .	60
II.6	Les outils associés . . . . .	61
II.6.1	Analyse en Composantes Principales (PCA) . . . . .	62
II.6.2	Quantification Vectorielle ( <i>Vector Quantizer VQ</i> ) . . . . .	63

II.6.3	Clustering . . . . .	65
II.7	Les outils utilisés . . . . .	66
II.8	Conclusion . . . . .	66
<b>III</b>	<b>SVM et Fiabilité</b>	<b>69</b>
III.1	Introduction . . . . .	71
III.2	Fil conducteur . . . . .	71
III.2.1	La fonction d'état-limite . . . . .	71
III.2.2	La représentation des SVM . . . . .	73
III.3	La méthode de J.E. Hurtado - SVM(H) . . . . .	74
III.3.1	Le principe de l'algorithme . . . . .	74
III.3.2	Le détail pas à pas [Hur04a] . . . . .	74
III.4	Discussion et propositions d'améliorations de l'algorithme . . . . .	77
III.4.1	La fonction noyau . . . . .	78
III.4.2	Le plan d'expériences initial - carré latin . . . . .	79
III.4.3	Choix des points d'apprentissage dans la marge . . . . .	83
III.4.4	Approche multi-échelle . . . . .	85
III.4.5	Bilan : trois phases d'apprentissage . . . . .	87
III.4.6	Critères de validation . . . . .	88
III.5	Bilan sur l'algorithme SMART . . . . .	93
III.5.1	Une vision globale . . . . .	94
III.5.2	Une frontière explicite . . . . .	95
III.5.3	Paramétrage de la méthode . . . . .	95
III.5.4	En conclusion . . . . .	95
III.6	Subsets Simulations et Classification : méthode <sup>2</sup> SMART . . . . .	96
III.6.1	<i>Subsets simulations</i> . . . . .	96
III.6.2	Algorithme SVM et <i>subsets simulations</i> - <sup>2</sup> SMART . . . . .	104
III.7	Conclusion . . . . .	113
<b>IV</b>	<b>Validation</b>	<b>115</b>
IV.1	Introduction . . . . .	117
IV.2	Outils de développement . . . . .	117
IV.3	Les critères de validation . . . . .	117
IV.4	Description des exemples . . . . .	118
IV.4.1	Validation de l'apprentissage - la méthode SMART . . . . .	118
IV.4.2	Validation de la méthode <sup>2</sup> SMART . . . . .	120



IV.5	Paramètres et suivi de la méthode d'apprentissage . . . . .	122
IV.5.1	Les paramètres de la méthode SMART . . . . .	122
IV.5.2	Le suivi de la méthode SMART . . . . .	123
IV.5.3	Discussion . . . . .	129
IV.6	Validation de la méthode <sup>2</sup> SMART . . . . .	130
IV.6.1	Analyse de formes complexes . . . . .	130
IV.6.2	Précision et stabilité du résultat . . . . .	131
IV.6.3	Influence de la dimension . . . . .	132
IV.7	Conclusion . . . . .	133
<b>V</b>	<b>Régression et analyse de sensibilité</b>	<b>135</b>
V.1	Introduction . . . . .	137
V.2	SVM et régression : quels apports pour la fiabilité ? . . . . .	137
V.2.1	Classification ou régression ? . . . . .	137
V.2.2	Optimiser la régression par SVM . . . . .	138
V.3	Les produits de l'analyse de sensibilité . . . . .	139
V.3.1	Les moments statistiques . . . . .	140
V.3.2	Quelques données théoriques sur les indices de Sobol' . . . . .	140
V.3.3	L'évaluation des indices de Sobol' . . . . .	144
V.3.4	Conclusion . . . . .	147
V.4	Estimation des moments statistiques et indices de Sobol' . . . . .	147
V.4.1	Méthode employée . . . . .	147
V.4.2	Création de la base de données . . . . .	148
V.4.3	Intervalle de confiance . . . . .	148
V.4.4	La poutre en flexion . . . . .	149
V.4.5	Treillis . . . . .	151
V.4.6	Discussion . . . . .	153
V.5	Méthode Sparse Grid . . . . .	155
V.5.1	L'interpolation par la méthode de la grille . . . . .	155
V.5.2	Optimisation de la grille . . . . .	157
V.5.3	Premiers résultats . . . . .	157
V.5.4	Discussion . . . . .	158
V.6	Conclusion . . . . .	159

<b>VI Applications Mécaniques</b>	<b>161</b>
VI.1 Introduction . . . . .	163
VI.2 Exemples simples de la mécanique . . . . .	164
VI.2.1 Une sphère sous pression . . . . .	164
VI.2.2 Un treillis . . . . .	168
VI.2.3 Un oscillateur à 2 degrés de liberté primaire-secondaire . . . . .	169
VI.2.4 Discussion . . . . .	174
VI.3 Plaque trouée . . . . .	175
VI.3.1 Description du modèle mécanique . . . . .	175
VI.3.2 Résultats du calcul fiabiliste . . . . .	176
VI.3.3 Plaque à 8 trous . . . . .	177
VI.4 Le projet PROSIR . . . . .	182
VI.4.1 Description du modèle mécanique . . . . .	184
VI.4.2 Résultats fiabilistes . . . . .	185
VI.4.3 Le défaut aléatoire . . . . .	187
VI.5 Conclusion . . . . .	188
<b>Conclusions et perspectives</b>	<b>189</b>
<b>Bibliographie</b>	<b>200</b>
<b>Table des figures</b>	<b>203</b>
<b>Liste des tableaux</b>	<b>206</b>
<b>Annexes</b>	<b>207</b>
Annexe 1. Les méthodes à noyaux . . . . .	209
Annexe 2. SQP . . . . .	211
Annexe 3. SVM et internet . . . . .	215



# Introduction

**L**ES MÉTHODES de calcul mécano-fiabilistes sont désormais largement connues et maîtrisées dans quelques laboratoires de mécanique intégrant le calcul scientifique et l'optimisation pour la conception des structures. La prise en compte de l'incertain dans les analyses mécaniques est en effet une condition indispensable pour un dimensionnement optimal et robuste.

Coupler modèles mécaniques et données incertaines permet d'étudier d'une part la fiabilité de composants ou de systèmes et, d'autre part, l'influence de la variabilité des paramètres sur leur comportement. Cependant, les modèles mécaniques sont de plus en plus complexes suivant en cela l'approfondissement des connaissances (comportement non-linéaire, dynamique, fatigue, mécanique de la rupture, etc). Ils nécessitent, pour leur étude, l'utilisation d'outils numériques très souvent construits sur la méthode des éléments finis (MEF). Les méthodes de couplage mécano-fiabiliste se doivent alors d'intégrer de telles modélisations pour rendre efficace et exploitable les études de fiabilité au delà des cas académiques. Cette nécessaire complexité rend les analyses de plus en plus exigeantes en temps de calcul et il s'agit alors de mettre en œuvre celles qui demandent un minimum d'appels au calcul mécanique tout en garantissant la robustesse des résultats visés.

Le but de nos travaux est donc de faire passer de telles études du stade académique au stade industriel en couplant les modèles stochastiques aux modèles mécaniques les plus avancés. Pour atteindre cet objectif, il est nécessaire de créer des procédures robustes et efficaces démontrant leur validité et tout leur intérêt dans une approche probabiliste. En effet, la conception mécano-fiabiliste tente de maîtriser un risque qu'elle explicite, et donc induit l'impossibilité d'un risque zéro. Cette vision probabiliste n'est pas toujours appréciée des concepteurs déterministes qu'elle peut déranger dans leurs certitudes. Elle présente pourtant un avantage certain, celui d'associer à un dimensionnement donné une probabilité de défaillance et, souvent, sa sensibilité

par rapport aux paramètres de conception.

La problématique à résoudre est le passage de la théorie probabiliste en fiabilité mécanique à une pratique qui la rende accessible à des problèmes industriels, caractérisés par leur dimension et leur complexité. Ce stade est partiellement acquis mais quelques verrous subsistent pour les problèmes de dimension importante (de 20 à 50 variables) et également pour des cas complexes en terme d'état-limite non régulier. Lorsque le calcul est rapide, il est facile de s'appuyer sur des résultats de simulation. Mais pour des calculs mettant en jeu d'importantes ressources numériques, il faut économiser impérativement le temps de calcul.

L'enjeu est donc de proposer une méthode efficace et souple. Efficace pour économiser le temps de calcul et souple pour permettre son utilisation dans la majorité des problèmes de la mécanique, sans restreindre le domaine de validité à telle ou telle approche, tout en le maîtrisant. Il s'agit d'organiser la démarche fiabiliste et de proposer de nouvelles approches afin de garantir l'obtention de résultats validés avec le minimum d'expériences numériques. L'aspect non intrusif de la méthode proposée est également primordiale de notre point de vue. Cette caractéristique est une des contraintes d'adaptabilité importante pour une intégration aisée avec les modèles numériques ou analytiques actuellement utilisés par les concepteurs.

Ce document est découpé en 6 chapitres. Ce découpage important reflète la démarche scientifique de la thèse.

Le premier chapitre, méthodes de la fiabilité, décrit les méthodes utilisées et leurs évolutions. Il permet la compréhension des techniques utilisées et l'identification de leurs points faibles. En conclusion de l'analyse, la substitution de modèle apparaît comme une solution pertinente, en particulier via l'apprentissage statistique.

Le deuxième chapitre présente les outils utilisés : les Machines à Vecteurs Supports (SVM : *Support Vector Machine*) et leurs applications.

Le troisième chapitre présente l'utilisation de l'apprentissage pour le calcul de la fiabilité. Le développement d'une méthode de fiabilité à partir des SVM et d'une approche en classification constitue le cœur de la thèse.

Le chapitre suivant présente la validation des solutions proposées sur des cas académiques.

L'analyse de la sensibilité, deuxième volet des approches probabilistes, ne peut

échapper à notre intérêt même si notre principal objectif reste l'analyse de fiabilité. C'est l'objet du cinquième chapitre.

Le dernier chapitre de la thèse met en application les méthodes développées sur des exemples de la mécanique.



## **Chapitre I**

# **Méthodes de calcul de la fiabilité des structures**

Ce premier chapitre présente la résolution du problème de fiabilité et les méthodes couramment utilisées de nos jours. La fin du chapitre donne les grandes lignes de la démarche suivie pour ce travail de recherche.



## Sommaire

---

<b>I.1</b>	<b>Introduction</b>	<b>7</b>
<b>I.2</b>	<b>Un problème de fiabilité</b>	<b>8</b>
I.2.1	La fiabilité en mécanique...	8
I.2.2	...une coopération entre divers acteurs	8
<b>I.3</b>	<b>Le calcul mécano-fiabiliste</b>	<b>10</b>
I.3.1	La fonction d'état-limite	10
I.3.2	La mesure d'une probabilité	10
I.3.3	Deux familles de méthodes	11
I.3.4	Substitution de l'état-limite	12
<b>I.4</b>	<b>Transformation des variables de base</b>	<b>12</b>
<b>I.5</b>	<b>FORM et SORM</b>	<b>13</b>
I.5.1	Le concept	13
I.5.2	Approximation de Taylor	13
I.5.3	La recherche du point de conception	14
I.5.4	Le calcul des dérivées	15
I.5.5	Les limites du calcul FORM/SORM	16
<b>I.6</b>	<b>Améliorer le calcul des gradients</b>	<b>16</b>
I.6.1	Méthode de différentiation directe	17
I.6.2	Implémentation de la DDM	18
I.6.3	Approximant de Padé [Wik07d]	21
<b>I.7</b>	<b>Les méthodes de simulation</b>	<b>22</b>
I.7.1	<i>Crude</i> Monte-Carlo	22
I.7.2	Simulations d'importance	24
I.7.3	Simulations directionnelles	25
I.7.4	Carré Latin	25
I.7.5	Subset Simulations	26
I.7.6	Conclusion	27
<b>I.8</b>	<b>Substituer la fonction de performance</b>	<b>27</b>
I.8.1	Pourquoi substituer l'état-limite?	27
I.8.2	Surfaces de réponse polynomiales	28
I.8.3	Réseaux de neurones ( <i>Neural Network</i> NN)	31
I.8.4	Classification et régression par les SVM	33
<b>I.9</b>	<b>Bilan</b>	<b>34</b>

---

## I.1 Introduction

**L**A PRISE en compte de l'incertain dans les analyses mécaniques est nécessaire pour un dimensionnement optimal et robuste des structures. C'est avec cet objectif que les approches probabilistes pour la mécanique des structures sont développées depuis plusieurs décennies. Ces méthodes permettent en effet d'étudier d'une part la fiabilité de composants ou de systèmes, et d'autre part l'influence de la variabilité des paramètres sur le comportement du composant ou du système.

Les modèles mécaniques sont souvent complexes (comportement non-linéaire, dynamique, fatigue, mécanique de la rupture, etc) nécessitant, pour leurs résolutions, l'utilisation de codes éléments finis. Cette complexité doit être prise en compte par les méthodes de couplage mécano-fiabiliste si l'on veut que l'analyse de fiabilité présente un sens. Cette prise en compte requiert un effort supplémentaire sur les ressources informatiques nécessaires. En effet, les méthodes de couplage nécessitent un appel au calcul déterministe du modèle mécanique pour des tirages judicieux de variables aléatoires. Ce nombre d'appels augmente généralement avec le nombre de variables aléatoires, ainsi le temps de calcul devient rapidement prohibitif par les méthodes traditionnelles.

L'objectif principal est donc de minimiser le nombre de calculs à effectuer tout en conservant une robustesse et une confiance satisfaisante du résultat : la probabilité de défaillance et ses facteurs d'importance.

Ce chapitre présente les principes de base du calcul fiabiliste ainsi que les différentes méthodes utilisées aujourd'hui. Il met en évidence les difficultés de mise en œuvre d'une approche couplant mécanique et statistique. Il aboutit à proposer des pistes de travail autour de récentes avancées dans le domaine.

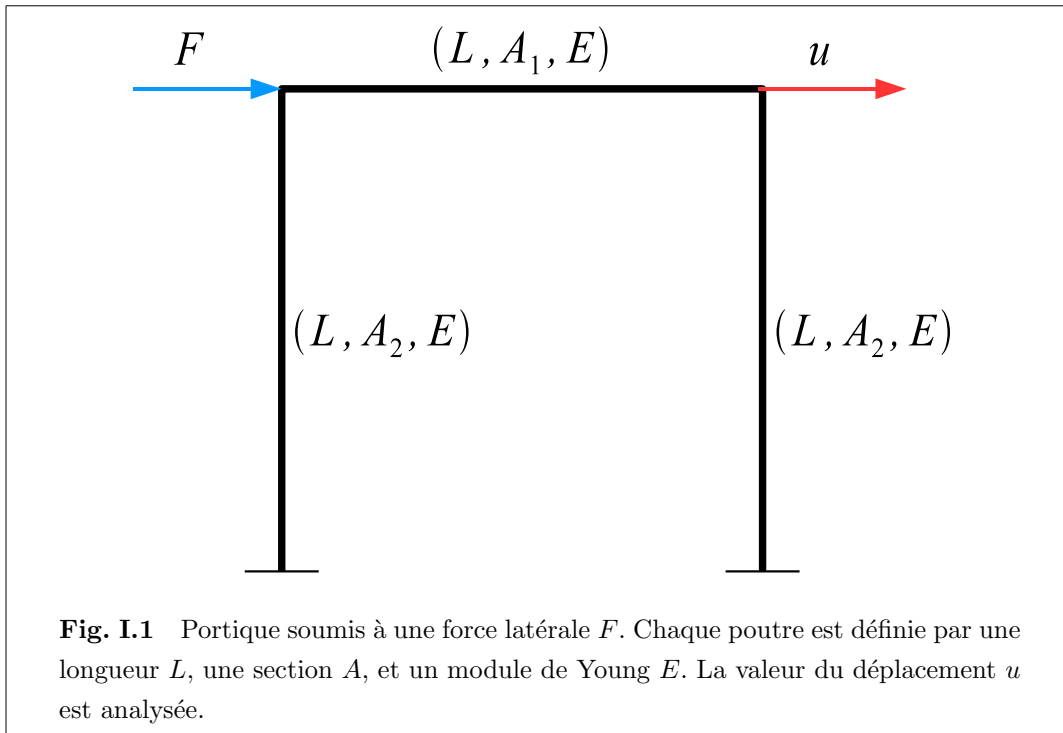
Les rappels et critiques sur les méthodes actuelles sont issus d'articles de la littérature et d'auteurs tels que : K. Breitung [BH89], O. Ditlevsen [DM05], M. Lemaire [Lem08], H.O. Madsen [MKL86], concernant les généralités, et J.B. Menzies [Men99] et R. Rackwitz [Rac01], pour quelques regards sur les évolutions.

## I.2 Un problème de fiabilité

Concrètement, quelles sont les compétences et procédures mises en jeu lors d'une étude de fiabilité ? Un exemple très simple permet ici de mettre en exergue les différentes étapes d'un calcul de fiabilité.

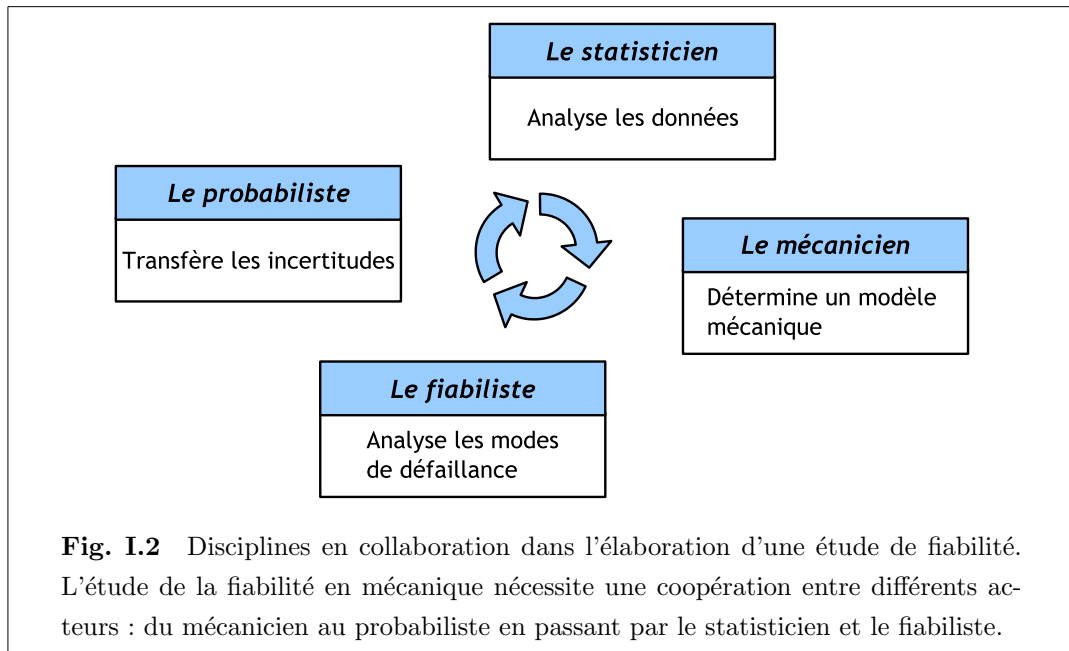
### I.2.1 La fiabilité en mécanique...

Considérons un exemple simple issu de la mécanique : la tenue d'un portique soumis à une force latérale. La figure I.1 présente la structure. Les poutres du portique sont définies par une longueur  $L$ , des sections  $A_1, A_2, A_3$ , et un module de Young  $E$  aléatoires. Pour vérifier la tenue de cette structure le déplacement  $u$  est analysé.



### I.2.2 ...une coopération entre divers acteurs

Une étude de fiabilité requiert, sur le même scénario, divers points de vue de différentes disciplines [Lem04]. L'expert en fiabilité analyse et décrit le scénario de défaillance ; le mécanicien propose un modèle pour évaluer l'événement défini ; ensuite le statisticien étudie les données ; enfin, le probabiliste propose une méthode de calcul



pour transférer les incertitudes. Ce schéma n'est pas une suite logique d'actions, mais une étude en collaboration des différentes disciplines. Le travail de thèse s'inscrit dans la dernière phase. Il s'agit de proposer de nouvelles méthodes d'évaluation de la probabilité de défaillance associée à un scénario dans le cas de problèmes complexes en grandes dimensions.

Si ce schéma, illustré en figure I.2 est appliqué à l'exemple du portique :

- le fiabiliste décrit le scénario de défaillance considéré : il s'agit d'un déplacement horizontal  $u$  supérieur au déplacement critique  $u_{limite}$  ;
- le mécanicien fournit un modèle issu de la théorie des poutres qui permet de calculer le déplacement  $u$  en fonction des paramètres géométriques et des caractéristiques matériau ;
- le statisticien étudie les dispersions des données géométriques et des caractéristiques matériau à partir de données d'essais et/ou de rapports d'experts ;
- le probabiliste va réunir modèles et données afin d'évaluer la probabilité que le déplacement dépasse le seuil critique, c'est l'étude de fiabilité. Il pourra également mesurer la dispersion du déplacement maximal et évaluer les paramètres qui influencent cette dispersion, c'est l'étude de sensibilité.

### 1.3 Le calcul mécano-fiabiliste

Le calcul mécano-fiabiliste intègre des notions de mesures de probabilités. Sa mise en œuvre nécessite de mettre en place un certain formalisme mathématique. Cette section décrit la mise en place de la fonction d'état-limite, des variables aléatoires et les grandes familles de méthodes permettant de résoudre un problème de fiabilité.

#### 1.3.1 La fonction d'état-limite

Soit  $X$  le vecteur des *variables de base*, qui peuvent être des variables définissant les actions extérieures, ou les caractéristiques matériau, ou encore les caractéristiques géométriques d'un composant ou d'un système mécanique. L'aléa sur ces variables est défini par la densité conjointe de probabilité. Ce vecteur aléatoire est défini dans l'espace *physique*.

Une *fonction de performance* ou *fonction d'état-limite*  $G(X)$  est définie telle que  $G(X) > 0$  soit le *domaine de sûreté*, et  $G(X) \leq 0$  soit le *domaine de défaillance*, ceci pour un scénario de défaillance donné. La frontière entre les deux domaines, *i.e.*  $G(X) = 0$ , est appelée l'*état-limite*. Pour des éléments de structures, la fonction de performance est assez souvent définie comme suit :

$$G(X) = R_{critique} - S(X) \quad (I.1)$$

où  $S(X)$  est la réponse du système appelée *sollicitation* et  $R_{critique}$  est la limite critique de la réponse étudiée appelée *résistance*.

#### 1.3.2 La mesure d'une probabilité

L'analyse de fiabilité a pour objectif l'estimation de la probabilité d'occurrence de défaillance due à la variabilité des variables de base, pour un scénario de défaillance donné. Pour le scénario défini à l'équation I.1,  $P_f$  représente la probabilité que la réponse du système  $S(X)$  excède la limite critique  $R_{critique}$ . Cette probabilité peut s'obtenir par intégration directe de la *fonction de densité conjointe*  $f_X(x)$  sur le domaine de défaillance :

$$P_f = \int_{G \leq 0} f_{X_1, \dots, X_n}(x_1, \dots, x_n) dx_1 \dots dx_n \quad (I.2)$$

où  $f_{X_1, \dots, X_n}(x_1, \dots, x_n)$  est la densité de probabilité conjointe des variables aléatoires de base  $x_i$ .

Les probabilités étudiées étant des probabilités de défaillance de systèmes mécaniques généralement fiables, leur ordre de grandeur est par conséquent très petit. Le calcul de l'intégrale est alors, sauf cas académique, impossible à calculer : l'erreur d'intégration étant de l'ordre de l'intégrale à calculer. La plupart des méthodes développées pour la fiabilité cherche, pour cette raison, à estimer cette probabilité de défaillance sans passer par le calcul de l'intégrale.

### I.3.3 Deux familles de méthodes

Pour résoudre le problème de fiabilité, deux types de méthodes se distinguent.

- La première famille est basée sur l'étude de la fonction de performance  $G(X)$ . La probabilité de défaillance est définie de la manière suivante :

$$P_f = P\{G \leq 0\} \quad (\text{I.3})$$

Pour le calcul de cette probabilité, il n'est pas nécessaire de connaître la forme de l'état-limite. Cette famille comprend les méthodes de simulation : principalement la simulation de Monte-Carlo et simulation selon le carré latin, (*Latin Hypercube* [Wik07c]). Des échantillons de la population des variables de base sont générés pour obtenir une population représentative sur laquelle une analyse statistique peut être conduite.

- La seconde famille est basée sur l'étude de l'état-limite. Ce qui importe, c'est la forme de la frontière  $G(X) = 0$ . Le but étant d'estimer l'intégrale suivante :

$$P_f = \int_{G \leq 0} f_X(x) dx \quad (\text{I.4})$$

L'étude de l'état-limite permet non seulement de calculer la probabilité de défaillance, mais également de définir d'autres indicateurs comme les *indices de fiabilité* ou encore le *point de conception* appelé aussi par abus de langage *point de défaillance le plus probable*. La recherche de ce point permet d'obtenir des informations supplémentaires très utiles à l'analyse des structures : *les facteurs d'importance* liés au point de conception. Ces indicateurs ne sont pas disponibles lors de l'utilisation de méthodes basées sur la fonction de performance. Deux catégories se distinguent :

- les méthodes basées sur le développement de Taylor de l'état-limite au voisinage du point de conception : FORM (*First Order Reliability Method*), pour un développement de Taylor au premier ordre, et SORM (*Second Or-*

*der Reliability Method*), pour un développement au second ordre. Ces méthodes nécessitent la connaissance de la valeur de l'état-limite ainsi que de ses gradients par rapport aux variables de base dans le voisinage du point de conception. La méthode SORM nécessite elle, de déterminer les dérivées du second ordre.

- Les méthodes utilisant la théorie statistique, c'est-à-dire les dérivées des simulations de Monte-Carlo. Ces dernières comprennent les méthodes orientées sur une description représentative de l'ensemble de la réponse : simulations d'importance (*Importance Sampling*), simulations directionnelles (*Directional Sampling*).

### 1.3.4 Substitution de l'état-limite

Toutes les méthodes de calcul font généralement appel à un solveur numérique (le code éléments finis) pour obtenir une évaluation de la réponse, ou à un substitut adéquat de la fonction de performance. Dans le cas de structures complexes, la fonction de performance est définie de manière implicite par l'intermédiaire d'un code éléments finis (ou autre solveur de ce type). Dans ce cas, il y a un intérêt non négligeable à approximer de manière simple mais "cependant précise" la fonction d'état-limite, afin d'éviter un nombre d'appels trop important au code de calcul.

La construction d'un approximant de la fonction d'état-limite constitue, ces dernières années, une des voies de développement des méthodes de la fiabilité. Les méthodes telles que les surfaces de réponse (*Response Surface Method - RSM*) relèvent de cette optique. D'autres méthodes basées sur ce même principe sont également utilisées par les équipes de recherche, par exemple des méthodes utilisant les techniques et outils issus de l'Apprentissage Statistique (*Statistical Learning*) tels que les réseaux de neurones (*Neural Networks*), ou encore plus récemment, les machines à vecteurs supports (*Support Vector Machine - SVM*).

Les avantages et inconvénients de ces différentes méthodes sont exposés dans la suite, ainsi que les voies d'amélioration sur lesquelles le travail de thèse s'oriente.

## 1.4 Transformation des variables de base

Pour la plupart des méthodes de calcul, le préalable au calcul de la probabilité de défaillance consiste à se placer, non pas dans l'espace des variables physiques, mais dans un espace dit *standard*. Le vecteur des variables de base  $X$ , défini par

sa fonction de densité conjointe est transformé en un vecteur de variables normales centrées, normées et indépendantes  $U$ , défini dans l'*espace standard*. Les transformations classiques utilisées sont celles de Nataf et Rosenblatt. La transformation de Nataf consiste en particulier à passer de variables aléatoires  $x_i$  de lois marginales quelconques et éventuellement corrélées à des lois normales centrées réduites non corrélées (indépendantes)  $u_i$ , par l'expression suivante :

$$u_i = T_i(x_k) = \Gamma_{ij} \Phi^{-1} \left[ F_{X_j}(x_j) \right] \quad (\text{I.5})$$

où  $\Gamma_{ij}$  est l'inverse de la triangularisation de Choleski de la matrice de corrélation équivalente,  $F_{X_j}(x_j)$  est la fonction de répartition de la variable  $x_j$  et  $\Phi^{-1}$  est l'inverse de la loi de répartition gaussienne standard. Une méthode de calcul de la transformation est disponible dans [DKP86], où des détails sont donnés sur un calcul approché de la matrice de corrélation équivalente.

## I.5 FORM et SORM

### I.5.1 Le concept

Ces méthodes sont considérées comme les plus efficaces en terme de rapport entre temps de calcul et précision de la réponse. Elles sont les plus utilisées et conviennent dans une majorité des cas.

La première étape consiste à rechercher le point de défaillance le plus probable  $P^*$ , encore appelé point de conception, dans l'espace des variables standard. Ensuite, la fonction d'état-limite est approximée par son développement de Taylor au premier ordre (FORM) ou au second ordre (SORM), autour du point de conception. Ces 2 méthodes d'approximation sont illustrées sur la figure I.3.

### I.5.2 Approximation de Taylor

Dans l'espace standard, on note  $H(U)$  la fonction d'état-limite (qui était notée  $G(X)$  dans l'espace des variables physiques  $X$ ). Le développement de Taylor au second ordre au point  $U^*$  s'écrit comme suit :

$$H(U) \approx H(U^*) + a^T(U - U^*) + \frac{1}{2}(U - U^*)^T H_e(U - U^*) \quad (\text{I.6})$$

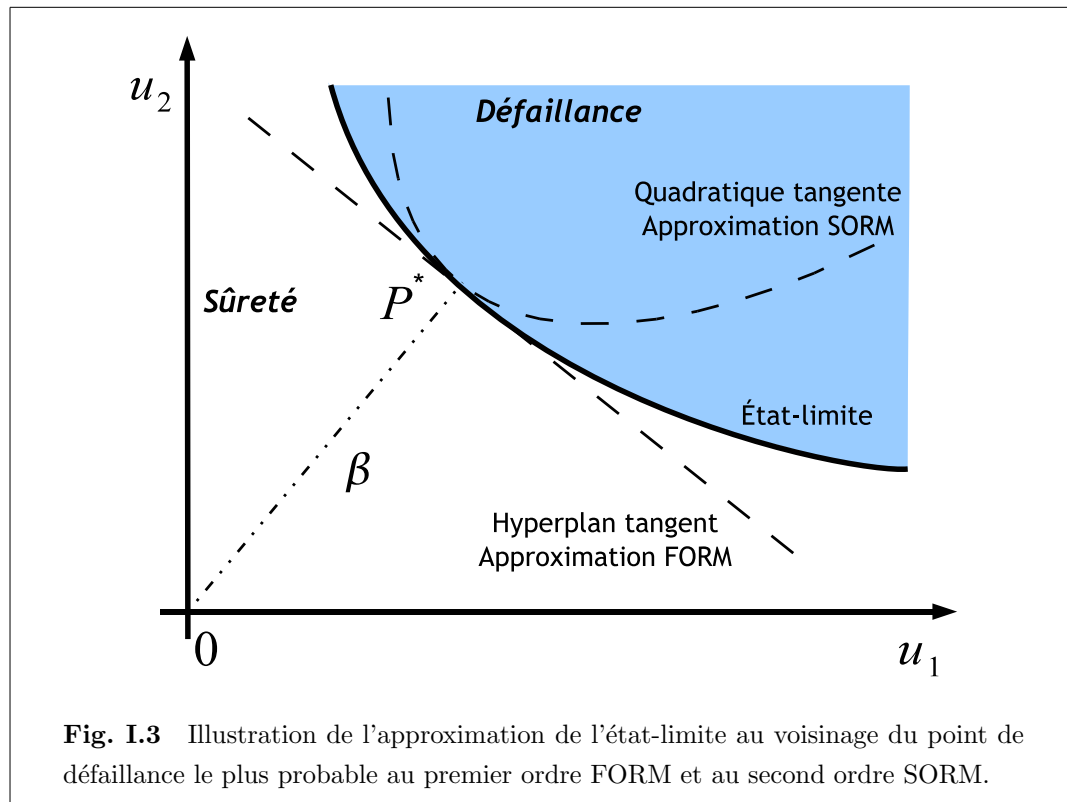


où  $a$  est le vecteur des gradients évalués au point de développement, et  $H_e$  le Hessien :

$$\begin{aligned} a &= \nabla H|_{U^*} \\ \mathbf{H}e_{ij} &= \left. \frac{\partial^2 H}{\partial u_i \partial u_j} \right|_{U^*} \end{aligned} \quad (\text{I.7})$$

Pour la méthode FORM, le terme du Hessien n'est pas conservé.

Il faut déterminer le point du développement  $P^*$  puis évaluer les grandeurs  $a$  et  $H_e$  le cas échéant.



**Fig. I.3** Illustration de l'approximation de l'état-limite au voisinage du point de défaillance le plus probable au premier ordre FORM et au second ordre SORM.

### I.5.3 La recherche du point de conception

Le point de conception  $U^*$  est solution du problème d'optimisation suivant :

$$\text{minimiser } \beta = \sqrt{U^T U} \text{ sous la contrainte } H(U) = 0 \quad (\text{I.8})$$

Cela correspond à chercher la distance minimale entre l'état-limite et l'origine dans l'espace standard. Ce problème d'optimisation est résolu par des méthodes itératives utilisant les gradients pour l'obtention de la direction de descente.

La résolution du problème donne une valeur de  $\beta$  qui correspond à l'indice de fiabilité dit de Hasofer et Lind :

$$\beta = -\alpha^T U^* \quad (\text{I.9})$$

où :

$$\alpha = \frac{\nabla H(U^*)}{|\nabla H(U^*)|} \quad (\text{I.10})$$

est le vecteur normal à l'hyperplan tangent à l'état-limite au point de conception, hyperplan tangent dont l'équation est la suivante :

$$\frac{H(U)}{|\nabla H(U^*)|} = \alpha^T (U - U^*) = \beta + \alpha^T U \quad (\text{I.11})$$

La méthode SORM utilise le même principe. Au point de conception, l'état-limite est approximé par une fonction du second ordre qui peut s'écrire comme suit :

$$\frac{H(U)}{|\nabla H(U^*)|} = w^T U + \beta + \frac{1}{2} (U - U^*)^T B (U - U^*) \quad (\text{I.12})$$

où :

$$B_{ij} = \frac{1}{|\nabla H(U^*)|} H_{eij}|_{U^*} \quad (\text{I.13})$$

#### I.5.4 Le calcul des dérivées

La méthode FORM, approximation par un hyperplan, nécessite le calcul des gradients de la fonction de performance par rapport à l'ensemble des variables de base. Comme la fonction de performance n'est généralement connue que de manière implicite, l'obtention de ses gradients se fait généralement par différences finies. Cette stratégie nécessite au moins  $n + 1$  appels au code de calcul par itération de l'algorithme d'optimisation (nombre d'itérations : de l'ordre de 10). L'approche SORM est d'un coût de calcul encore plus sévère, puisqu'elle requiert le calcul du Hessian par différences finies (uniquement en  $P^*$ ).

Par conséquent, pour utiliser ces méthodes sur des problèmes plus complexes en terme de nombre de variables de base, il faut développer de nouvelles approches. L'utilisation de la différentiation directe en est une, voir section I.6. Elle permet, à partir d'une seule résolution, d'obtenir les gradients directement. Cependant, cela nécessite d'implémenter l'opérateur de différentiation à toutes les étapes de l'algorithme de calcul, donc de construire des codes dédiés à l'étude des sensibilités. Cela permet également une meilleure convergence des algorithmes, puisque la précision du calcul des gradients par différences finies n'est pas optimale. Cette précision joue en effet un rôle important dans la convergence vers le point de conception.

### 1.5.5 Les limites du calcul FORM/SORM

Il faut souligner deux principales difficultés liées aux méthodes d'approximation FORM/SORM. La première difficulté tient dans le calcul des dérivées, en particulier le réglage des différences finies. L'autre point concerne l'éventuelle présence de points de conception multiples, Der Kiureghian [DKD98].

Le problème des dérivées, abordé dans la sous-section précédente, 1.5.4, est lié à l'utilisation des différences finies. D'une part, cette technique nécessite un nombre de calculs importants en cas de problèmes de grande dimension. D'autre part, ce calcul est dépendant du pas de différences finies, et peut devenir parfois très instable en fonction de la forme de la fonction étudiée.

Le second point est commun à la plupart des techniques d'optimisation. Il s'agit de l'arrêt de l'algorithme sur des minima locaux. En effet, l'optimisation par les gradients ne permet pas l'obtention d'un extremum global. Se pose alors la question de l'unicité de la solution. Dans le cadre d'une étude de fiabilité, la non unicité de la solution équivaut à la présence de points de défaillance multiples. La non prise en compte de points multiples peut entraîner une erreur plus ou moins importante sur l'évaluation de la probabilité de défaillance.

Par conséquent, ces méthodes sont régulièrement critiquées quant à la validité des résultats dans des cas pathologiques. Cependant, un grand nombre de problèmes rencontrés en pratique peuvent être résolus par ces méthodes, et le compromis temps de calcul-robustesse joue aujourd'hui en leur faveur.

La méthode SORM possède les mêmes inconvénients que la méthode FORM. Le principal étant la recherche du point de conception dont l'unicité n'est jamais assurée. Une fois le point  $P^*$  obtenu, le calcul de la dérivée seconde peut s'avérer très fastidieux en grande dimension. Rackwitz présente également des limites par rapport à la dimension de l'espace dans [Rac01].

## 1.6 Améliorer le calcul des gradients

Une constante dans la résolution du problème de la fiabilité réside dans le problème d'optimisation permettant l'obtention des points de défaillance. La plupart des algorithmes analytiques utilise les gradients de la fonction de performance pour calculer les directions de descente. Nous faisons ici un aparté sur ce sujet, et sur la manière d'améliorer ce calcul.

Pour obtenir des méthodes analytiques d'approximation plus fiables et plus rapides, il faut améliorer les différentes étapes de l'algorithme. Le passage le plus délicat et le plus important se situe au niveau du calcul des gradients. Pour des raisons de complexité des modèles numériques utilisés, le calcul de dérivées du modèle par rapport à une variable de conception se fait par différences finies. Cette méthode est très simple à implémenter sur une procédure quelconque de résolution numérique. Cependant, elle comporte des inconvénients non négligeables. Son coût en terme de temps de calcul est assez prohibitif. Elle demande en effet un grand nombre d'appels à la résolution du problème mécanique initial pour obtenir les résultats. De plus, le pas de calcul doit être ajusté pour éviter des erreurs importantes sur le calcul de sensibilités. De telles erreurs détériorent le processus de convergence des algorithmes d'optimisation et donc leur efficacité. En effet, le calcul des gradients par différences finies cumule les erreurs de la réponse des éléments finis sur un minimum de 2 calculs déterministes. Le résultat est ainsi très sensible aux erreurs de discrétisation du modèle éléments finis.

La détermination de la sensibilité d'un modèle mécanique par rapport à ses variables d'entrée est de plus en plus utilisée pour de multiples raisons. Les sensibilités peuvent avoir diverses applications telles que : l'obtention d'indicateurs d'importance des variables qui guident le concepteur dans une phase de conception, la détermination de directions de recherche dans les procédures d'optimisation... Il est important que le calcul de ces sensibilités, ou gradients, soit réalisé de manière précise et fiable par rapport au calcul de la réponse du système, ceci pour le bon déroulement des algorithmes d'optimisation et pour l'obtention d'une information non erronée sur le système étudié.

### I.6.1 Méthode de différentiation directe

La méthode des différences finies, utilisée par le plus grand nombre pour sa facilité de mise en place sur les codes de calcul existants, ne permet de garantir ni précision, ni stabilité pour le calcul des sensibilités. D'autres méthodes sont pourtant disponibles, telles que la méthode de différentiation directe : *Direct Differentiation Method (DDM)*. L'objectif est d'utiliser au maximum la différentiation directe, qui réduit l'ordre de grandeur de l'erreur sur le gradient à celui de l'erreur sur le calcul mécanique.

Cette méthode existe dans la littérature de fiabilité, dans celle de l'optimisation et dans les études de sensibilité des modèles. Des exemples dans le domaine du di-

mensionnement de structures sont présentés par K.K. Choï [CD00], S.H. Kim [KH02], D.T. Nguyen [NQR98] ou encore R. Serban [SF96]. Ils comparent l'utilisation de la méthode des différences finies à celle de la différentiation directe pour l'évaluation des sensibilités et pour l'optimisation. Les résultats en terme de précision et de stabilité sont comparés à ceux obtenus par différences finies, et les travaux présentés mettent en évidence le gain obtenu en appliquant la DDM.

### I.6.2 Implémentation de la DDM

L'implémentation de la DDM permet :

- d'une part, un gain de temps important en terme de temps de calcul, car à l'aide d'une unique résolution déterministe (certes plus longue qu'un simple calcul) sont obtenues les dérivations par rapport à toutes les variables de conception ;
- d'autre part, elle est très bénéfique en terme de précision : en effet la précision du calcul est proche de la précision du calcul éléments finis.

La difficulté tient dans la formulation de la méthode de dérivation directe. La méthode de calcul des sensibilités dépend du type de problème considéré et de la méthode de résolution utilisée, et elle doit être implémentée au niveau de la résolution du problème déterministe, par exemple au niveau de la résolution par éléments finis. La mise en place de ce type de calcul nécessite alors de "revisiter" le code de calcul. L'aspect intrusif de la DDM est visible ici. En effet, il faut que l'opérateur de différentiation soit présent à chaque opération de la résolution EF. On distingue la *méthode directe* et la *méthode adjointe*. Pour comprendre la différence entre ces deux méthodes, il est nécessaire de revenir sur les équations de base de la différentiation directe.

#### Les équations de la DDM

Le but est d'obtenir les sensibilités d'une fonction  $f$  par rapport aux variables de conception  $x_n$ . Une telle fonction peut également dépendre de variables d'état du système  $y_i$ .

$$f = f(x_n, y_i) \quad (\text{I.14})$$

Pour  $x_n$  donné, la solution des équations qui gouvernent le système est un vecteur  $y_i$ . Ces équations régissant le système sont contenues dans :

$$R_k(x_n, y_i(x_n)) = 0 \quad (\text{I.15})$$

Les variables d'état peuvent dépendre explicitement ou implicitement des variables de conception. Si le système d'équations est non linéaire, alors la résolution s'effectue de manière incrémentale en amenant les résidus  $R_k$  à zéro.

La sensibilité de  $f$  est :

$$\frac{df}{dx_n} = \frac{\partial f}{\partial x_n} + \frac{\partial f}{\partial y_i} \frac{dy_i}{dx_n} \quad (\text{I.16})$$

La dérivée partielle,  $\frac{\partial f}{\partial x_n}$ , peut-être évaluée en variant le dénominateur et en ré-évaluant le numérateur. La dérivée totale nécessite de résoudre les équations du problème. Dans l'expression précédente, seule l'évaluation de  $\frac{dy_i}{dx_n}$  n'est pas triviale.

Les équations gouvernant le système doivent être toujours satisfaites, ainsi la dérivée totale de l'équation doit être nulle par rapport à toutes les variables de conception. Soit :

$$\frac{dR_k}{dx_n} = \frac{\partial R_k}{\partial x_n} + \frac{\partial R_k}{\partial y_i} \frac{dy_i}{dx_n} = 0 \quad (\text{I.17})$$

ce qui s'écrit encore :

$$\frac{\partial R_k}{\partial y_i} \frac{dy_i}{dx_n} = -\frac{\partial R_k}{\partial x_n} \quad (\text{I.18})$$

Ainsi il est possible d'évaluer  $\frac{dy_i}{dx_n}$  et d'en substituer le résultat dans la dérivée totale de  $f$  :

$$\frac{df}{dx_n} = \frac{\partial f}{\partial x_n} - \frac{\partial f}{\partial y_i} \left[ \frac{\partial R_k}{\partial y_i} \right]^{-1} \frac{\partial R_k}{\partial x_n} \quad (\text{I.19})$$

On note :

$$\varphi_k = -\frac{\partial f}{\partial y_i} \left[ \frac{\partial R_k}{\partial y_i} \right]^{-1} \quad (\text{I.20})$$

### La différentiation directe

La méthode qui consiste à évaluer d'abord  $\frac{dy_i}{dx_n}$ , puis à le substituer dans l'expression de la dérivée totale est appelée la *méthode directe*. Résoudre  $\frac{dy_i}{dx_n}$  nécessite de résoudre l'équation I.18 pour chacune des variables de conception. Pour une variation d'une variable de conception, seul le membre de droite change. Ainsi, lorsque la matrice  $\frac{\partial R_k}{\partial y_i}$  peut être explicitement factorisée et conservée, il est alors simple et peu coûteux de résoudre cette équation pour différents membres de droite. Pour des systèmes non linéaires, itératifs, cette matrice n'est jamais factorisée explicitement. La multiplication des variables de conception peut rendre coûteuse l'utilisation de la méthode directe.

### La méthode adjointe

La *méthode adjointe* représente alors une alternative intéressante pour palier l'inconvénient précédemment cité. Le principe est de résoudre d'abord l'équation adjointe :

$$\frac{\partial R_k}{\partial y_i} \varphi_k = - \frac{\partial f}{\partial y_i} \quad (\text{I.21})$$

$\varphi_k$  est appelé le vecteur adjoint et est substitué dans l'équation I.19 pour obtenir la sensibilité totale. Le vecteur adjoint ne dépend pas des variables de conception mais de la fonction  $f$ .

Le concepteur peut donc choisir entre ces deux méthodes de calcul, en fonction des caractéristiques de son problème, celle qui est la moins coûteuse.

### La différentiation automatique

L'utilisation systématique de la différentiation directe analytique nécessite un travail d'intrusion dans le code. Une méthode plus simple d'implémentation a fait son apparition : la *différentiation automatique*, ou *différentiation algorithmique*. Cette méthode est basée sur l'application de la différentiation en chaîne, G. Corliss [Cor92] en propose une bibliographie. P. Cusdin, dans [CM03b] et [CM03a], utilise cette méthode dans le cadre de la dynamique des fluides et en explique les principes de base.

Ce type de méthode permet d'automatiser le processus de dérivation sans écrire les formules de différentiation dans le code de calcul. La différentiation automatique s'oppose ainsi à la différentiation directe que l'on pourrait qualifier d'analytique dans laquelle on formule véritablement les sensibilités. Cette méthode s'implémente à l'aide de logiciels dédiés, qui vont dériver à la chaîne toutes les opérations basiques réalisées par le code de calcul. C'est une méthode très utile pour des problèmes complexes et des codes complexes où l'écriture de la différentiation directe est fastidieuse. La thèse réalisé par T. Haukaas, [Hau03], met en évidence la difficulté d'évaluer les sensibilités d'une quantité (dépendantes des variables d'état du système) par rapport aux variables de design pour différentes méthodes de résolution, différents types d'éléments, différents calculs...

En terme de précision, la différentiation automatique est très fiable, et également rapide. Cette méthode est un bon compromis entre la méthode directe et les différences finies. De nombreux logiciels permettent de réaliser de tels calculs, pour en citer quelques uns : ADIFOR, TAMC, DAFOR, ADOL... Cependant, **il est néces-**

**saire d'avoir les sources du solveur** pour mettre en place ce type d'outils. Il existe deux manières d'implémenter ce type d'algorithme :

- soit par transformation du code source : tout le code de calcul est transformé par l'algorithme et de nouvelles lignes de code sont insérées pour le calcul des dérivées ;
- soit en utilisant des données dérivées. Dans ce cas, une nouvelle structure de données est créée contenant chaque valeur et sa dérivée, et les opérateurs sont redéfinis pour travailler sur ces nouvelles structures de données.

Cette méthode permet également d'obtenir, en utilisant les algorithmes adéquats, les dérivées d'ordres supérieurs.

Certains logiciels de calcul, tels qu'ANSYS, ABAQUS, Code ASTER, commencent à proposer le calcul de sensibilités.

### I.6.3 Approximant de Padé [Wik07d]

La différentiation directe permet l'introduction des méthodes d'ordres supérieurs. P. Guillaume en est un précurseur dans le domaine de l'optimisation de forme utilisant les dérivées d'ordres successifs [GHR97]. Le principe consiste à évaluer au point de calcul (point  $P^*$  à une certaine itération du processus, par exemple), les dérivées d'ordres successifs, et d'obtenir l'approximation de l'état-limite au voisinage de ce point, par le polynôme de Taylor à un degré supérieur qui est déterminé par l'algorithme. Cette approximation fournit les sensibilités de l'état-limite par rapport aux variables, et permet d'obtenir un indice de fiabilité FORM ou SORM de la fonction de performance étudiée.

Dans le même esprit, la recherche de l'approximant de Padé de l'état-limite est possible. L'intérêt de celui-ci est d'obtenir une approximation de l'état-limite qui est valable sur un voisinage plus grand autour du point de calcul. L'approximation est meilleure et ainsi l'optimisation plus rapide et plus sûre. L'inconvénient réside dans le calcul des sensibilités par la différentiation directe. Mais la construction se fait également à partir des dérivées d'ordres successifs de la fonction de performance par rapport aux variables de design. La théorie de l'approximant de Padé et son utilisation pour la mécanique sont détaillées dans plusieurs publications de P. Guillaume : [Gui97b, Gui97a, GH00, GM94]. L'obtention d'une telle approximation revient à l'obtention d'une surface de réponse au voisinage du point, mis à part que sa construction n'est possible qu'à l'aide d'une unique résolution du modèle mécanique et sans construction d'un plan d'expériences déterministe, puisque les coefficients sont obtenus



nus par différentiation automatique.

Des exemples d'utilisations de l'approximant de Padé dans ANSYS sont disponibles dans les rapports techniques suivants : [CAD99], [TBBA99].

Cette méthode présente l'avantage d'être plus précise que l'approximation de l'état-limite par un hyperplan, et également d'être rapide car l'utilisation de la différentiation directe est incluse dans la méthode. La DDM est l'unique manière d'obtenir les dérivées d'ordres supérieurs. L'utilisation de cette méthode nécessite les outils nécessaires à la mise en place d'une DDM dans le problème à résoudre.

## I.7 Les méthodes de simulation

Les méthodes de simulation sont basées sur le calcul de la fonction de performance pour un échantillon représentatif (et souvent de grande taille) de la population. Ces méthodes sont les plus coûteuses en temps de calcul, mais restent la référence sur le résultat en probabilité de défaillance. Certaines variantes permettent de diminuer les nombres d'appels à la fonction de performance. Le temps de calcul est ainsi diminué, mais reste trop important pour s'en contenter dans le cadre de calculs industriels, dans lesquels le temps de calcul de la fonction de performance est non négligeable et le nombre de variables aléatoires important. De multiples travaux sont toutefois réalisés sur ces méthodes, car elles sont robustes et qu'elles peuvent être utilisées sur des substituts de la fonction de performance (cf. section I.8) où l'évaluation de l'état-limite devient alors très rapide.

Après quelques données sur la simulation de Monte-Carlo originale (*crude* MC), les variantes les plus courantes sont détaillées. Les illustrations sont inspirées du manuel de Phimeca Software©[PHI06] et de M. Lemaire [Lem04].

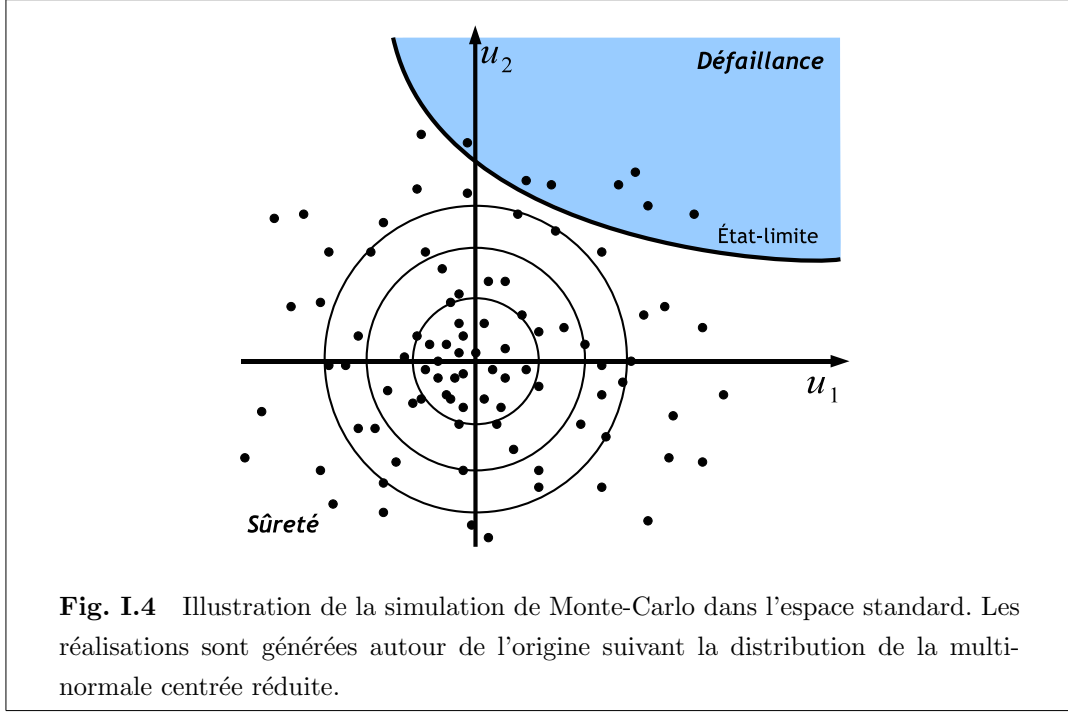
### I.7.1 Crude Monte-Carlo

Le but est d'estimer l'intégrale suivante :

$$P_f = \int_{G \leq 0} f_X(x) dx \quad (\text{I.22})$$

Cette intégrale limitée au domaine de défaillance peut être ramenée à une intégrale sur l'espace entier en introduisant la fonction indicatrice :

$$\gamma_f(x) = \begin{cases} 1 & \text{si } G(x) \leq 0 \\ 0 & \text{si } G(x) > 0 \end{cases} \quad (\text{I.23})$$



L'expression de la probabilité de défaillance prend alors la forme suivante :

$$P_f = \int \gamma_f(x) f_X(x) dx \quad (\text{I.24})$$

La probabilité de défaillance est alors estimée par l'expression suivante :

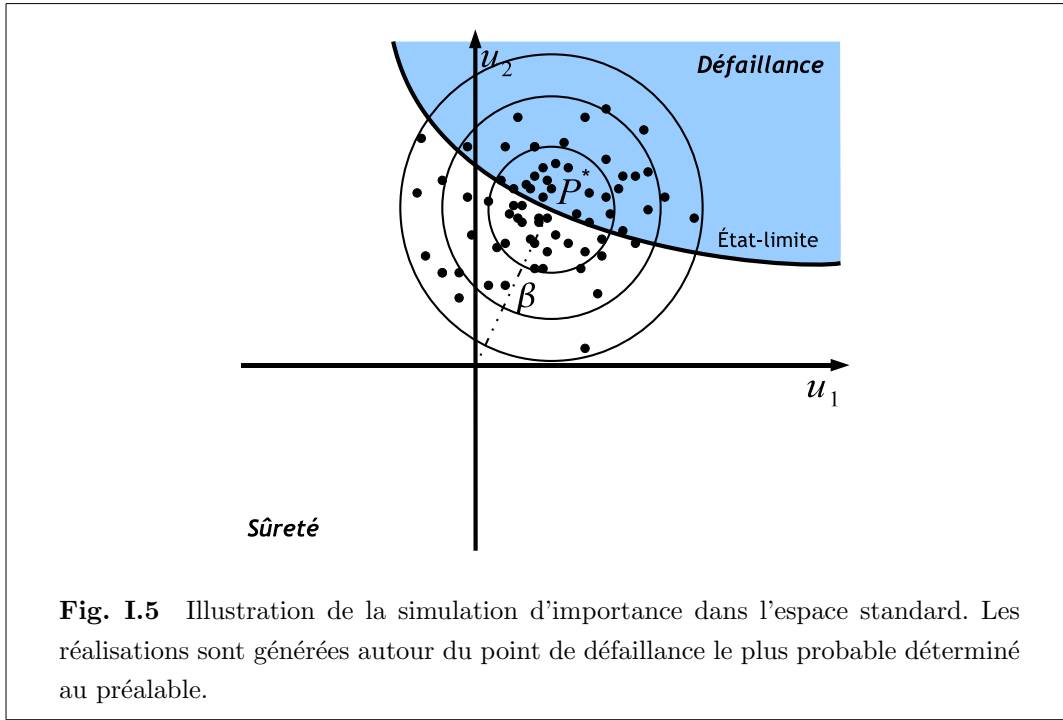
$$\hat{P}_f = \frac{1}{n} \sum_{i=1}^n \gamma_f(x_i) = E[\gamma_f(x)] \quad (\text{I.25})$$

en utilisant  $n$  réalisations du vecteur  $X$ , sachant que pour obtenir un bon estimateur de la probabilité de défaillance, il faut un grand nombre  $n$  de simulations. Dans le cas de la fiabilité des structures, ceci est d'autant plus vrai que les probabilités recherchées sont faibles (les structures sont fiables). Pour estimer une probabilité de défaillance de l'ordre de  $10^{-n}$ , il faut classiquement un échantillon de  $10^{n+2}$  voire  $10^{n+3}$ . Cette méthode est extrêmement coûteuse en temps de calcul. Elle reste cependant l'une des plus robustes.

Les simulations de Monte-Carlo sont considérées comme les réponses de référence au problème de la fiabilité. Il faut préciser que seuls les tirages dans le domaine de défaillance contribuent à l'amélioration de l'estimateur de la probabilité de défaillance. Les variantes de la méthode basique de Monte-Carlo utilisent cette propriété et tentent de rendre les tirages plus performants. L'objectif est de réduire la variance

de l'estimation de la probabilité de défaillance, à nombre de simulations identiques, ou de réduire le nombre de simulations pour atteindre une variance donnée sur cette probabilité de défaillance.

### 1.7.2 Simulations d'importance



Le principe est de produire les échantillons dans les régions qui contribuent le plus à la mesure de l'intégrale. Les points ayant le plus de poids dans la probabilité de défaillance sont généralement situés autour du point de conception si celui-ci est unique. Par conséquent, l'essentiel des tirages est situé autour de  $P^*$ .

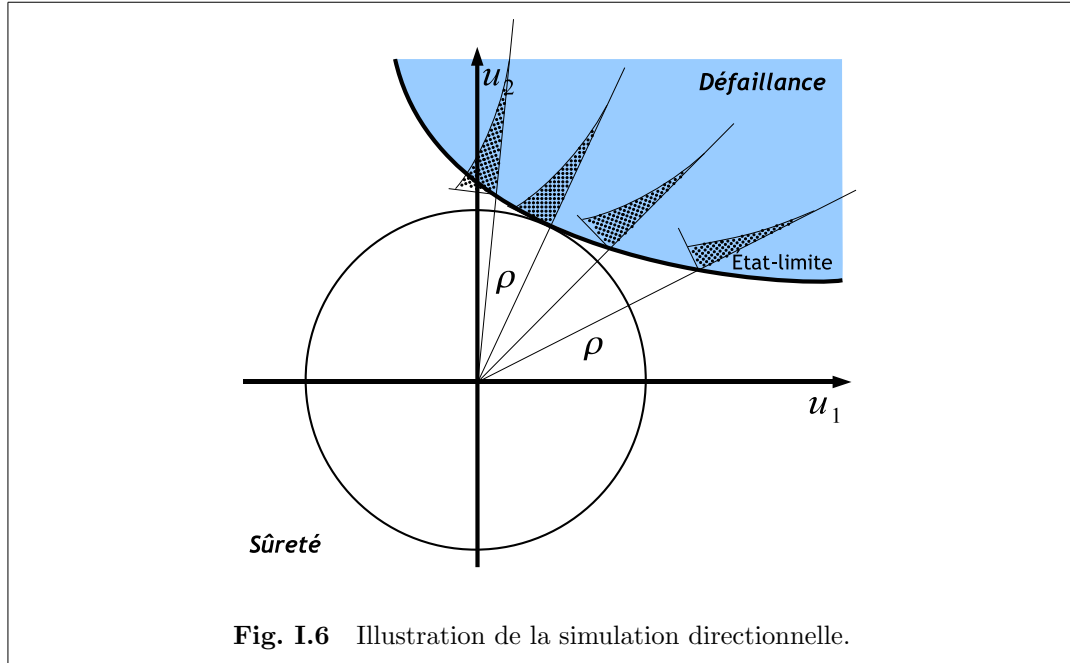
Deux étapes sont nécessaires : il s'agit de chercher d'abord le point de défaillance le plus probable et de réaliser ensuite une simulation centrée sur ce point.

La probabilité de défaillance est alors estimée par l'expression suivante :

$$\hat{P}_f = \frac{1}{n} \sum_{i=1}^n \gamma_f(x_i) \frac{\varphi_n(u_i)}{f_n(u_i - u^*)} \quad (\text{I.26})$$

où  $\varphi_n$  est la densité multi-normale et  $f_n$  dépend du choix de la densité utilisée pour la simulation.

### I.7.3 Simulations directionnelles



Les simulations directionnelles sont basées sur le concept de probabilité conditionnelle et utilisent la symétrie de l'espace standard. Les simulations sont alors réalisées de manière radiale. Des directions dans l'espace sont généralement générées aléatoirement et la probabilité de défaillance est calculée conditionnellement à la direction.

La probabilité de défaillance est alors estimée par l'expression suivante :

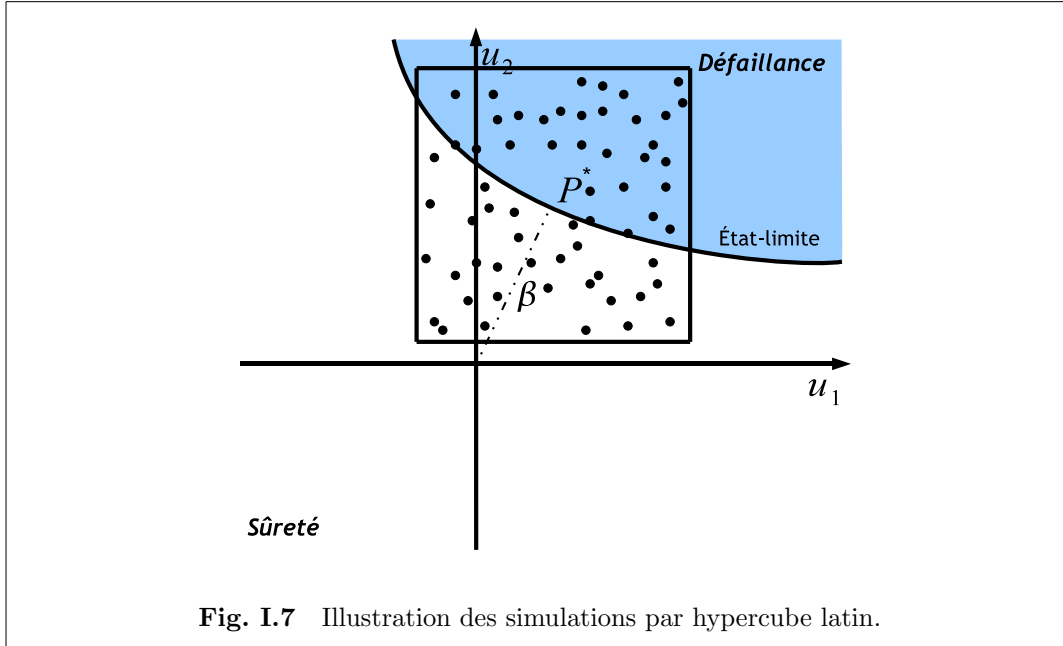
$$\hat{P}_f = \frac{1}{n} \sum_{i=1}^n \left[ 1 - \chi_n^2(\rho_i)^2 \right] \quad (\text{I.27})$$

où  $\chi_n^2$  est la loi de répartition du  $\chi^2$  à  $n$  degré de liberté.

Il est nécessaire de déterminer les intersections des directions avec l'état-limite de manière itérative.

### I.7.4 Carré Latin

Dans cette méthode de simulation, les tirages sont effectués de manière uniforme dans un pavé autour du point de conception. La fonction de densité uniforme s'écrit :  $f_{U_k}(u_k) = \frac{1}{V(p)}$ , où  $V(p)$  est le volume du pavé.



**Fig. I.7** Illustration des simulations par hypercube latin.

La probabilité de défaillance s'estime par l'intégrale suivante :

$$\hat{P}_f = \frac{V(p)}{N} \sum_{i=1}^N \gamma_f(x_i) \varphi_n(u_i) \quad (\text{I.28})$$

### I.7.5 Subset Simulations

Une des variantes les plus récentes ou la plus récemment mise à jour est la méthode des *subsets simulations*. La version proposée par Au et Beck [AB01] permet de résoudre les problèmes de fiabilité dont la probabilité de défaillance est très faible, à un coût très intéressant par rapport au *crude* Monte-Carlo. Cette méthode est longuement décrite dans le chapitre III puisqu'elle sert de base aux développements proposés dans cette thèse.

De manière très succincte, l'idée consiste à définir la probabilité de défaillance comme un produit de probabilités conditionnelles, chacune de ces probabilités ne nécessitant qu'un nombre limité de simulations pour leurs évaluations respectives.

Cette méthode reste une méthode de simulation, dans le sens où elle nécessite un grand nombre d'évaluations de l'état-limite : de l'ordre de  $10000 \cdot n$  pour l'évaluation d'une probabilité de l'ordre de  $10^{-n}$ . De plus amples détails sont disponibles section III.6.1.

### I.7.6 Conclusion

Les avantages des méthodes de simulation sont de plusieurs ordres :

- il n’y a pas d’hypothèse établie sur la forme de l’état-limite ou sur la position du(des) point(s) de conception sur la linéarité ou la taille du problème ;
- l’erreur n’est pas affectée par la dimension du problème : le pourcentage d’erreur est très peu, voire pas du tout, sensible à la dimension du problème à l’inverse des autres méthodes approchées (type FORM).

Souvent, il est reproché aux méthodes de simulation l’absence d’information sur les points de défaillance. Hormis pour la simulation d’importance pour laquelle le point de défaillance est connu, il est possible d’utiliser des techniques de *clustering* des points de la zone de défaillance afin de déterminer le ou les points de conception. C’est ce que montre F. Perrin dans [PPS06].

L’utilisation de ces méthodes de simulations pour un problème de taille industrielle reste difficile en raison du coût. Il peut-être réduit en remplaçant la fonction d’état-limite par une approximation simple mais assez précise. Ainsi, il est possible de diminuer le temps de calcul en remplaçant un calcul éléments finis par la simple évaluation d’une fonction explicite. Cette solution est explorée dans la suite.

## I.8 Substituer la fonction de performance

Les récentes avancées des méthodes de couplage mécano-fiabiliste sont à mettre sur le compte des méthodes de substitution de modèles complexes. Ce ne sont pas des méthodes spécifiques de résolution du problème d’optimisation mécano fiabiliste, mais la substitution de la fonction de performance permet d’accélérer la résolution de problèmes de taille importante en utilisant les algorithmes existants. Des comparaisons de ces méthodes avec les classiques FORM et Monte-Carlo sont nombreuses dans la littérature. H.M. Gomes, dans [GA04], pense que ces méthodes permettront d’accélérer les procédures pour résoudre les problèmes de grande taille.

### I.8.1 Pourquoi substituer l’état-limite ?

Le remplacement de l’état-limite par un substitut présente plusieurs avantages :

- l’information nécessaire à la résolution par des méthodes analytiques (*i.e.* les valeurs, gradients et courbures de la fonction d’état-limite autour du point de conception), peut être fournie par un substitut de la fonction de performance ;

- pour les méthodes issues des simulations de Monte-Carlo, augmenter la précision nécessite de faire des appels supplémentaires de la fonction d'état-limite (résolutions du modèle éléments finis). Le temps de calcul peut être largement diminué par l'utilisation d'une **bonne approximation** de la fonction de performance ;
- enfin, ce type de méthode est moins affecté par l'augmentation de la dimension du problème, à l'inverse des méthodes d'approximation FORM/SORM.

Une grande partie des travaux recensés se sont orientés vers les surfaces de réponse (RSM : *Response Surface Methods*). Depuis quelques années, l'apprentissage statistique a fait son apparition chez les fiabilistes : réseaux de neurones et plus récemment les machines à vecteurs supports (SVM *Support Vector Machine*) sont utilisées pour remplacer l'appel à un solveur trop long.

### I.8.2 Surfaces de réponse polynomiales

L'utilisation de cette technique consiste à remplacer le calcul de la valeur de la fonction de performance par une expression analytique polynomiale. Le choix de la surface dépend du niveau d'approximation requis, mais surtout de la taille du problème.

#### Le principe

Très souvent, une forme quadratique est choisie pour l'écriture du substitut  $\hat{G}$  de la fonction de performance  $G$ . Autrement dit, le substitut s'écrit de la manière suivante :

$$\hat{G}(X) = c_0 + \sum_{i=1}^n c_i x_i + \sum_{i=1}^n c_{ii} x_i^2 \quad (\text{I.29})$$

ou, en incluant les termes croisés :

$$\hat{G}(X) = c_0 + \sum_{i=1}^n c_i x_i + \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_i x_j \quad (\text{I.30})$$

avec  $X$  le vecteur des  $n$  variables de base, et  $c_i$ ,  $c_{ij}$ , les coefficients du polynôme recherché. Les coefficients sont les inconnues du problème qu'il faut rechercher en utilisant un plan d'expériences de taille minimale. Le plan d'expériences peut être choisi minimum pour obtenir un nombre d'équations correspondant au nombre d'inconnues (mais l'interpolation n'est pas efficace), il peut également être plus complet et nécessiter l'utilisation des méthodes de régression pour le calcul des coefficients du polynôme.

### Mise en œuvre des surfaces de réponse et difficultés

Historiquement, dans le cadre de l'analyse de fiabilité, les surfaces de réponse sont utilisées pour approximer l'état-limite autour de la zone de présence du point de défaillance le plus probable  $P^*$ . L'objectif est d'évaluer le Hessien de manière efficace en approximant l'état-limite par une surface quadratique. Ainsi, il est possible d'obtenir l'indice de fiabilité SORM.

L'algorithme suit la chronologie suivante :

- recherche d'une surface autour du point de départ par une méthode de régression sur les points de calcul du plan d'expériences ;
- recherche du nouveau point de conception par la méthode FORM ou SORM sur la surface définie ;
- puis recherche d'une nouvelle surface autour du point de conception.

Ceci est itéré jusqu'à convergence de l'algorithme vers le point de défaillance le plus probable.

Les points délicats sont :

- le choix des polynômes : le degré, le type, etc ;
- la méthode de régression : pondérée ou non ;
- et bien entendu les points d'apprentissage qu'il faut déterminer.

En ce qui concerne le choix du méta-modèle, le développement à l'ordre 2 est souvent retenu. Un degré supérieur est souvent nécessaire pour une meilleure approximation de la fonction, mais cela devient coûteux dans des espaces de grande dimension. De plus, l'obtention d'un développement d'ordre 2 autour du point de défaillance le plus probable, permet l'évaluation immédiate de l'indice SORM.

La base de polynômes retenue est souvent la base la plus intuitive :  $[1, x, x^2, \dots, x^n]$ . Mais d'autres bases sont possibles et peuvent apporter quelques avantages. La base des polynômes d'Hermite est ainsi régulièrement utilisée pour ses propriétés d'orthogonalité par rapport à la mesure gaussienne. Souvent associé aux éléments finis stochastiques, le développement de la réponse sur le chaos polynomial d'Hermite est utilisé dans un cadre plus global d'analyse de sensibilités, et moins pour l'évaluation d'indices de fiabilité. En effet, le développement permet d'analyser principalement les tendances centrales de la réponse, voir les publications de B. Sudret et M. Berveiller [SBL04, Ber05].



### Développements sur le thème des surfaces de réponse

**(a) Le plan d'expériences.** Souvent, les expérimentations (calcul du résultat d'un tirage) sont réalisées aux points situés à la moyenne  $\pm k$  écart-type,  $k$  étant alors un paramètre déterminant pour le résultat obtenu. Comment choisir le plan d'expériences ? X.L. Guan, dans [GM01], montre que la variation du paramètre  $k$  influence de manière non négligeable sur les résultats du calcul de la fiabilité, le plan d'expériences est source d'instabilité dans la résolution du problème par les surfaces de réponses.

Par conséquent, les surfaces de réponse ne sont à utiliser qu'en connaissance de cause. Leur intérêt était d'obtenir des dérivées exactes pour le calcul de la descente, mais l'approximation tend à être instable. Pourtant la méthode plaît puisque de nombreux dérivés de l'algorithme de base existent dans la littérature, et de nombreuses recherches sont en cours sur les méthodes de régression qui permettent d'intégrer un calcul d'erreur.

Par exemple, I. Kaymaz utilise une méthode de régression pondérée dans [KM04]. Le principe est d'affecter un poids plus important aux points proches de l'état-limite, par exemple l'inverse de la distance.

**(b) Évaluer l'erreur ou l'instabilité.** La difficulté de la méthode reste le choix des points de calcul. Il s'agit de se rapprocher le plus rapidement possible de l'état-limite et du point de conception, car la zone de défaillance la plus probable possède les informations les plus importantes pour évaluer la fiabilité d'une structure par rapport à un scénario de défaillance. Dans cette optique, N. Gayton, dans [GBL03], propose une méthode qui tente d'optimiser les plans d'expériences. La méthode proposée permet de définir dans une première étape de l'algorithme, par un pré-processing de l'espace standard, une zone dans laquelle la recherche du point de conception se concentre.

Il propose également une méthode statistique originale d'évaluation du point de conception, à partir de l'ensemble des expérimentations réalisées. Il compare les  $n - 1$  points de conception obtenus en utilisant les  $n$  expérimentations moins une, prise tour à tour : *resampling* ou Jackknife. Le point de conception est considéré comme une variable aléatoire, laquelle est évaluée par l'obtention d'une distribution construite expérimentalement.

**(c) Une surface peu adaptative.** D. Roos [RB03] propose une méthode de surface adaptative, où une erreur est introduite et permet de contrôler la convergence de l'algorithme. Cette méthode n'utilise plus une représentation quadratique de la surface mais des fonctions non linéaires : elle est construite à l'aide de  $n$  simplexes. L'intérêt repose sur la non linéarité de la surface, et donc sur le fait que celle-ci est plus précise dans les cas d'un manque de continuité ou d'un état-limite complexe. Cette méthode s'apparente plus aux réseaux de neurones.

**(d) Limites.** Le problème des surfaces de réponse réside dans la rigidité de la surface *a priori* que l'on impose à la forme de l'état-limite. Dans certains cas, s'évertuer à utiliser une quadrique ne conduit pas à une précision suffisante.

Une autre limite réside dans le temps de calcul, car cette méthode nécessite un nombre d'appels à l'état-limite qui croît rapidement avec le nombre de variables de base. Cela rend cette méthode inutilisable pour des problèmes de taille importante.

Les méthodes issues de l'apprentissage statistique sont, de ce point de vue, plus souples, s'adaptent plus facilement à des cas de non linéarité importante de la fonction de performance, et sont moins sensibles à la dimension du problème.

### I.8.3 Réseaux de neurones (Neural Network NN)

**(a) Le principe.** Cette méthode consiste à utiliser un réseau de neurones afin de simuler l'état-limite. L'intérêt est de pouvoir utiliser les algorithmes de fiabilité existant sur le réseau plutôt que sur le modèle mécanique. Ceci est possible une fois l'apprentissage du réseau effectué.

Ces méthodes sont très utiles lorsque les méthodes conventionnelles FORM/RSM ne sont plus utilisables, si la dimension du problème est importante, par exemple. En effet, le temps de calcul de ces méthodes n'est pas proportionnel au nombre de variables du problème, comme peuvent l'être les méthodes plus conventionnelles.

Parmi les nombreux types de réseaux de neurones, certains ont été utilisés et donnent des résultats intéressants pour servir de substitut de la fonction de performance. Nous retiendrons les suivants : le *Multi Layer Perceptron* (MLP) et *Radial Basis Function Networks* (RBFN) qui sont les plus présents dans la littérature de la mécanique fiabiliste : voir M. Papadrakakis [PL02], P.C. Pendharkar [Pen02] ou J.E. Hurtado [HA01].

**(b) Le MLP.** Pour le premier type, MLP, le substitut prend la forme suivante :

$$\hat{G}(X) = \tilde{h} \left( \sum_{k=0}^m w_k h \left( \sum_{i=0}^n w_{ki} x_i \right) \right) \quad (\text{I.31})$$

où les  $w_k$  et  $w_{ki}$  sont des poids,  $x_i$  la  $i$ -ème coordonnées du vecteur  $X$ .  $h()$  et  $\tilde{h}()$  sont des fonctions en général non linéaires. La forme suivante peut être utilisée :

$$h(t) = \frac{1}{1 + \exp(-t)} \quad (\text{I.32})$$

Cette fonction est également utilisée pour  $\tilde{h}$  dans le but de faire de la classification. Pour la régression,  $\tilde{h} = t$  est plus souvent utilisée.

**(c) La RBFN.** La RBFN est basée non plus sur l'utilisation des coordonnées du vecteur aléatoire des variables de base mais sur un échantillon de réalisations de  $X$  :  $X_1, \dots, X_m$ , pour construire un modèle de la forme :

$$\hat{G}(X) = \sum_{i=1}^m w_i h(X, X_i) \quad (\text{I.33})$$

où les  $w_i$  sont des poids, et la fonction  $h$  est la gaussienne :

$$h(X, X_i) = \exp \left( -\frac{\|X - X_i\|^2}{2w^2} \right) \quad (\text{I.34})$$

La méthode des SVM utilise ce type de fonction mais les poids sont calculés de manière très différente (voir chapitre II). La fonction recherchée est du type :

$$\hat{G}(X) = \sum_{i=1}^m w_i K(X, X_i) - b \quad (\text{I.35})$$

où  $K(X, X_i)$  est une fonction noyau (Kernel function) et  $b$  un paramètre.

**(d) L'apprentissage.** Les algorithmes d'apprentissage de ces réseaux sont nombreux. Ils sont déterminants dans la recherche d'une fonction de substitution car ils vont conditionner l'obtention des différents paramètres comme les poids. Souvent, ces algorithmes utilisent les *Back Propagation techniques*.

Il s'agit en fait d'un problème d'optimisation : trouver les poids qui permettent de minimiser une fonction coût. Cette fonction permet le contrôle de la précision de l'approximation obtenue. L'optimisation est non-linéaire et peut aboutir à des minima locaux.

L'obtention d'une fonction analytique de substitution de la fonction de performance permet d'obtenir rapidement le résultat FORM ou SORM, ainsi que la probabilité de défaillance, mais également les sensibilités en dérivant les fonctions du réseau par un opérateur de dérivation à la chaîne dans le cas de réseau de plusieurs neurones [DGLQY05].

Les difficultés résident dans les algorithmes d'apprentissage, les moyens de contrôle de la précision ainsi que le choix de population d'apprentissage permettant de préparer le réseau. J.E. Hurtado compare différentes méthodes d'apprentissage dans [HA01].

#### I.8.4 Classification et régression par les SVM

Cette section présente les concepts des machines à vecteurs supports qui sont plus longuement détaillés dans le chapitre II.

En une phrase : les SVM permettent d'utiliser la souplesse des méta-modèles type réseaux de neurones en conservant une forte capacité de généralisation (pilotée par le choix des paramètres). Cet outil permet de réaliser des simulations performantes en utilisant la classification [Hur04a], mais peut également servir pour réaliser une régression permettant d'obtenir une approximation explicite de la fonction d'état-limite. À chacun des deux grands types d'applications de l'approche probabiliste, la fiabilité et la sensibilité, correspond une utilisation spécifique des propriétés de SVM : la classification ou la régression respectivement.

Ces outils statistiques sont intéressants pour plusieurs raisons :

- Ils permettent de travailler avec des populations restreintes, comme souvent en fiabilité. En effet, il est rare d'avoir un très grand nombre de données expérimentales et les données numériques sont parfois difficiles à obtenir (temps de calcul très longs).
- C'est un modèle qui est très flexible et permet une bonne approximation (en classification comme en régression).
- Pour la classification, Les SVM définissent une marge, dans laquelle se situe la frontière entre le domaine de défaillance et le domaine de sûreté, et qui permet de générer des nouveaux échantillons permettant d'affiner l'approximation.

J.E. Hurtado [Hur04a] présente la méthode des SVM ainsi que des outils statistiques permettant de réaliser d'autres gains de temps (cf. chapitre II). Les algorithmes utilisant les SVM diffèrent selon leur utilisation dans le cadre de la classification, ou celui de la régression. Pour la classification, le but est de séparer des échantillons se-

lon leur classe. Notre problème de fiabilité s'intègre parfaitement dans une optique de classification : il s'agit de séparer les échantillons selon leur appartenance au domaine de sûreté ou au domaine de défaillance. C'est un exercice de classification binaire. La régression s'apparente à une surface de réponse mais la méthode de régression est novatrice et utilise d'autres principes que les régressions utilisées par la RSM.

Cette piste est préférable aux réseaux de neurones d'après J.E. Hurtado [Hur04b] qui a également réalisé une étude comparative des réseaux de neurones [HA01].

## I.9 Bilan

L'objectif principal de nos travaux est de permettre la mise en place de la fiabilité pour des problèmes mécaniques complexes, cette complexité résidant au niveau du modèle mécanique ou du modèle stochastique. Il s'agit de permettre l'utilisation de ces approches, afin de prendre en compte les différents aléas pour une conception fiable.

Pour atteindre cet objectif, il faut rendre robuste le calcul de fiabilité mais également lui permettre d'intégrer plus de données aléatoires en accélérant les procédures. Différentes approches sont possibles car des actions à différents niveaux de l'analyse sont envisageables : amélioration des procédures d'optimisation, résolution du modèle mécanique, analyse des sensibilités (gradients), etc.

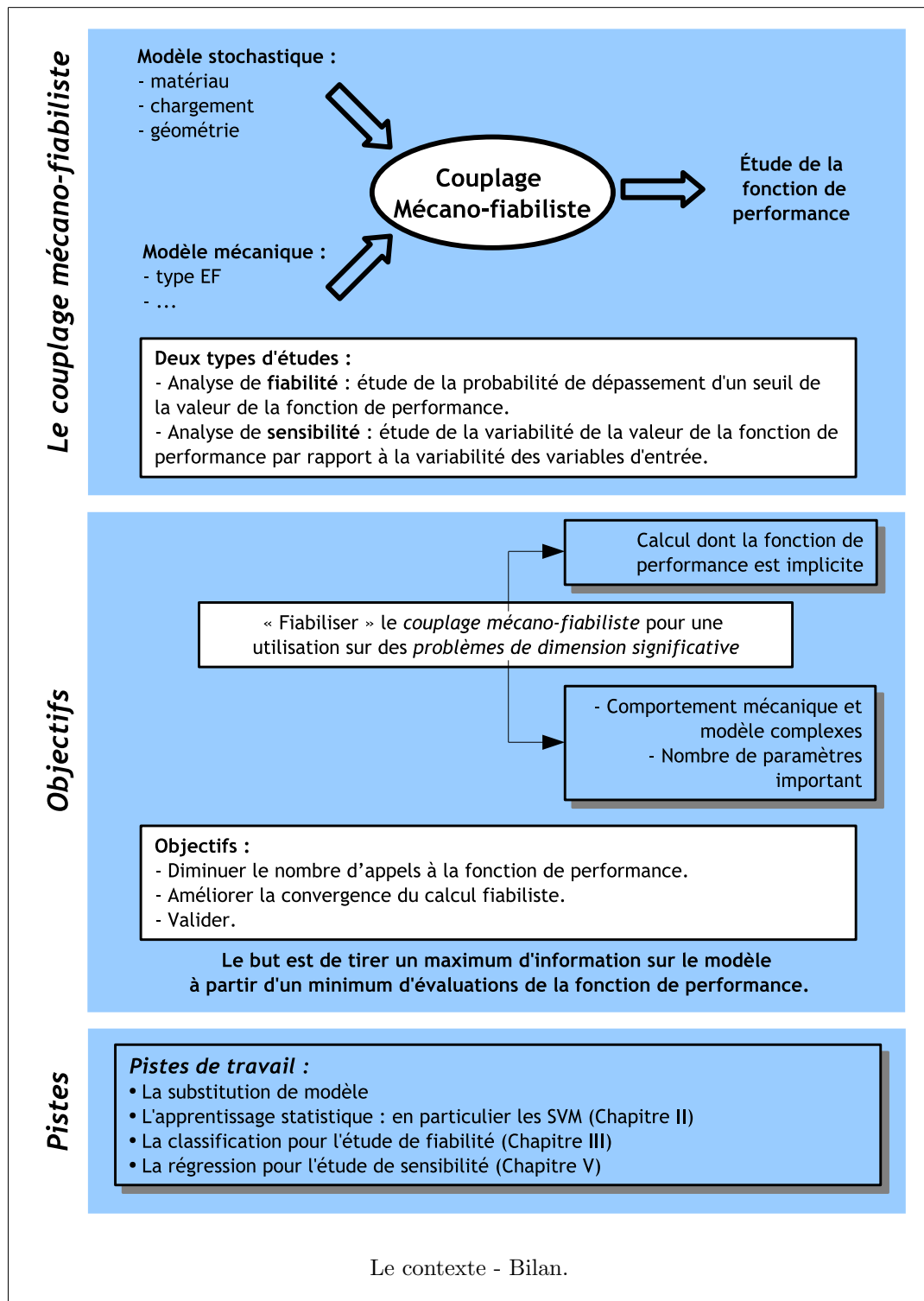
Différentes pistes ont été présentées dans ce chapitre, notamment la différentiation directe, les techniques de simulations innovantes, et les surfaces de réponse.

Cependant, **une approche non intrusive par rapport au modèle physique utilisé est indispensable** pour la généralisation souhaitée. Cette approche évite de remanier les codes de calculs actuels. Cette dernière condition est de notre point de vue primordiale pour l'application des méthodes fiabilistes au plus large éventail de domaines. Cette dernière priorité exclut d'emblée un recours aux méthodes de différentiation directe, malgré les avantages de celles-ci. Mais le développement de telles méthodes, motivé par l'optimisation, dans les codes ouvrira demain de nouvelles perspectives.

La substitution de l'état-limite semble alors la meilleure méthode pour permettre d'augmenter le nombre de variables aléatoires, ainsi que la complexité du problème à moindre coût. Ceci est dû au fait que le coût de l'optimisation dépend de manière moins forte de l'augmentation du nombre de variables. De plus, la substitution de l'état-limite permet de conserver les méthodes qui ont fait leurs preuves aujourd'hui :

les simulations ou les méthodes analytiques. L'apprentissage semble alors l'approche la plus souple et efficace, à condition de démontrer sa validité dans son domaine de définition.

Remarque sur la puissance de calcul : améliorer les temps de calcul peut passer par la parallélisation et la distribution des calculs pour les rendre utilisables sur cluster. Mais la parallélisation est un outil. Il s'agit bien sûr ici de trouver d'autres moyens, qui eux mêmes seront un jour parallélisés. Cependant, dans l'écriture de procédures, il est important de réfléchir à leur mise en œuvre sur des architectures de type cluster pour un calcul parallèle ou au moins distribué.



## Chapitre II

# Machines à Vecteurs Supports

Ce chapitre est consacré à la présentation de la théorie des machines à vecteurs supports (*Support Vector Machine SVM*). Les équations aboutissant à l'écriture du problème d'optimisation à résoudre pour l'obtention des paramètres SVM sont ici exposées.



---

## Sommaire

<b>II.1</b>	<b>Introduction</b>	<b>39</b>
<b>II.2</b>	<b>Classification et SVM</b>	<b>39</b>
II.2.1	Le cas linéaire	40
II.2.2	Le cas non linéaire - méthode à noyaux	43
II.2.3	Les classes non séparables	45
<b>II.3</b>	<b>Régression par SVM</b>	<b>47</b>
<b>II.4</b>	<b>SVM ou réseaux de neurones</b>	<b>49</b>
II.4.1	ERM et principe de régularisation	50
II.4.2	VC dimension et SRM	54
II.4.3	La grande dimension	57
<b>II.5</b>	<b>Optimisation des SVM</b>	<b>58</b>
II.5.1	Le choix des paramètres	58
II.5.2	Validation croisée	59
II.5.3	<i>Boosting</i> algorithm	60
<b>II.6</b>	<b>Les outils associés</b>	<b>61</b>
II.6.1	Analyse en Composantes Principales (PCA)	62
II.6.2	Quantification Vectorielle ( <i>Vector Quantizer VQ</i> )	63
II.6.3	Clustering	65
<b>II.7</b>	<b>Les outils utilisés</b>	<b>66</b>
<b>II.8</b>	<b>Conclusion</b>	<b>66</b>

---

## II.1 Introduction

LE PROJET de thèse est centré sur l'utilisation des outils de l'apprentissage statistique pour le développement de méthodes de fiabilité. Il est par conséquent nécessaire d'introduire le vocabulaire utile, ainsi que les outils et les techniques employés. Ce chapitre est donc consacré à une étude bibliographique des *machines à vecteurs supports* (en anglais : *Support Vector Machine*, d'où le sigle qui est utilisé SVM). D'autres outils régulièrement associés sont également présentés. Ces différents outils ne sont encore que très peu utilisés pour les études de fiabilité, dans lesquelles les méthodes de l'apprentissage statistique sont présentes principalement par les réseaux de neurones. Ces méthodes sont en revanche fortement développées pour des applications en traitement d'images, en classification des protéines, en reconnaissance de caractères, etc. Seul J.E. Hurtado [Hur04b], après avoir étudié et comparé les types de réseaux de neurones, s'est attardé sur les possibilités offertes par les SVM pour la fiabilité des structures.

Dans ce chapitre, le vocabulaire utilisé dans la théorie des SVM est introduit et les principes de base de la méthode sont expliqués. Ces différentes notions présentées s'appuient sur les travaux des auteurs de référence sur les SVM tels que Schölkopf, Cristianni, Smola, Vapnik [Vap98, SS03a, Sch00, CST00]. Les détails de ces principales références sont reprises par J.E. Hurtado sous l'angle fiabiliste dans [Hur04b].

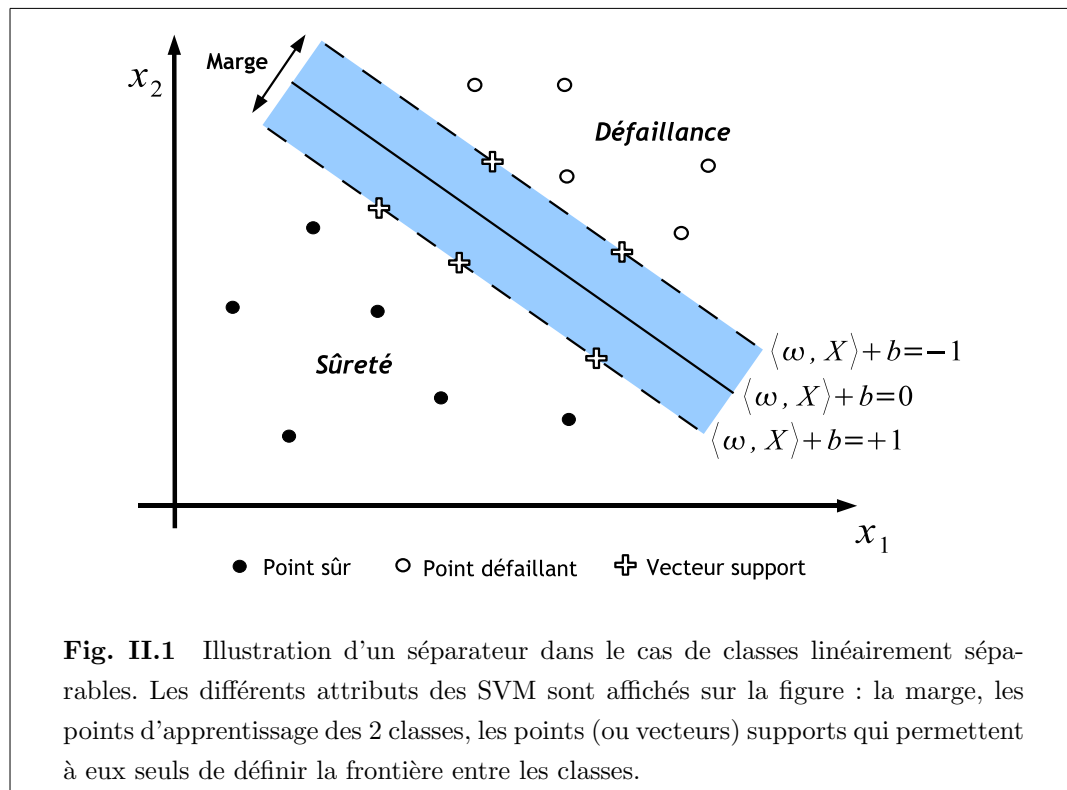
## II.2 Classification et SVM

Les SVM ont été créés pour résoudre les problèmes dits de *classification*. Le but est de séparer l'espace des données en classes. À chaque réalisation des variables de base est associée une classe. Le principe de l'algorithme est de créer un hyperplan qui sépare les données en deux classes (classification binaire), cf. figure II.1, en utilisant le principe de *marge maximale* : considérons des points d'apprentissage appartenant aux classes  $-1$  et  $+1$ , l'hyperplan maximisant la marge sépare les deux classes des points d'apprentissage de telle façon que la distance entre les points les plus proches de la marge et l'hyperplan soit maximisée. La première version de l'algorithme a

été proposée par Vladimir Vapnik [VL63] en 1963. Elle s'appliquait aux séparateurs linéaires (on parle de classes linéairement séparables). Puis, en 1992, Bernhard Boser, Isabelle Guyon et Vladimir Vapnik proposent la création d'un séparateur non linéaire en appliquant le “*Kernel trick*” (méthode à noyaux) au principe de la marge maximale, [BGV92]. L'algorithme de base n'est pas modifié, seul le produit scalaire classique est remplacé par une fonction noyau non linéaire. L'algorithme de marge maximale est alors utilisé dans un espace transformé appelé *espace support* (*feature space*). La transformation peut être non linéaire et l'espace support de très grande dimension : cela permet d'obtenir un séparateur qui est un hyperplan dans l'espace support, mais une fonction non linéaire dans l'espace de départ.

L'intérêt pour le problème de la fiabilité est simple : il s'agit d'associer une classe à la réponse du système mécanique pour la réalisation  $x$  des variables de base, classe “sûreté” (+1), et classe “défaillance” (−1). **Notre objectif est de trouver une formulation analytique de la frontière des classes.**

### II.2.1 Le cas linéaire



### Un problème de classification

Prenons tout d'abord le cas de classes linéairement séparables. Nous considérons deux classes  $+1$  et  $-1$ . Soit  $X_1, \dots, X_n$  un échantillon de  $n$  réalisations des variables de conception. Soit  $c_i \in \{-1, 1\}, i = 1, \dots, n$  les classes associées à chacune des réalisations. Nous cherchons un hyperplan qui sépare les deux classes de la forme :

$$G(X) = \langle \omega, X \rangle + b \quad (\text{II.1})$$

où  $\omega$  est un vecteur de paramètres, vecteur définissant la normale à l'hyperplan ; et  $b$  est appelé le seuil. La fonction de classification s'écrit alors :

$$c(X) = \text{sgn}(G(X)) = \text{sgn}(\langle \omega, X \rangle + b) \quad (\text{II.2})$$

On demande à cet hyperplan de maximiser la distance entre l'hyperplan et les deux classes, ce qui constitue une condition d'optimalité de position de l'hyperplan. Cela se comprend intuitivement sur la figure II.1. Le problème de classification se résume à déterminer le vecteur de paramètres  $\omega$  et le seuil  $b$ , en respectant les conditions d'optimalité de position de l'hyperplan.

### Le problème d'optimisation

Le problème d'optimisation à résoudre pour trouver les paramètres de l'hyperplan est alors le suivant :

$$\max_{\omega, b} \min_i \{ \|X - X_i\| \mid \langle \omega, X \rangle + b = 0, i = 1, \dots, n \} \quad (\text{II.3})$$

où  $\min_i \{ \|X - X_i\| \mid \langle \omega, X \rangle + b = 0, i = 1, \dots, n \}$  représente la distance minimale entre l'hyperplan et les points d'apprentissage.

L'écriture d'une classe par un signe ( $+1$  ou  $-1$ ) facilite l'écriture de l'optimisation.  $\gamma_i$  désigne la marge de chaque réalisation, marge qui s'écrit :

$$\gamma_i = c_i(\langle \omega, X_i \rangle + b), i = 1, \dots, n \quad (\text{II.4})$$

Le signe positif de  $\gamma_i$  montre une bonne classification. En effet, si le produit de  $c_i$  par  $\langle \omega, X_i \rangle + b$  est positif, cela signifie que le signe de  $c_i$  et le signe de l'évaluation du séparateur au point  $X_i$  sont identiques, et que le point  $X_i$  est classé du bon côté du séparateur.

Il y a une infinité de vecteurs  $\omega$  qui répondent au problème à un facteur multiplicateur près. Il s'agit alors d'imposer une condition supplémentaire pour obtenir

une solution unique. La condition suivante est imposée : la distance minimale entre l'hyperplan et la réalisation la plus proche est fixée, soit :

$$\min_{i=1,\dots,n} |\langle \omega, X_i \rangle + b| = 1 \quad (\text{II.5})$$

Cela implique que la marge soit égale à deux fois la distance minimale, soit :

$$\gamma = 2 \frac{|\langle \omega, X_i \rangle + b|}{\|\omega\|} = \frac{2}{\|\omega\|} \quad (\text{II.6})$$

Maximiser la marge  $\gamma$  correspond donc à minimiser la norme de  $\omega$ , ce qui conduit au problème d'optimisation suivant :

$$\min \frac{\|\omega\|^2}{2} \text{ sous la contrainte } c_i(\langle \omega, X_i \rangle + b) \geq 1, \quad i = 1 \dots n \quad (\text{II.7})$$

où les  $n$  contraintes imposent la bonne classification des points d'apprentissage par rapport au séparateur recherché.

### La résolution du problème d'optimisation

Pour résoudre le problème d'optimisation posé, on fait appel au Lagrangien, défini comme suit :

$$L(b, \omega, \alpha) = \frac{\|\omega\|^2}{2} - \sum_{i=1}^n \alpha_i [c_i(\langle \omega, X_i \rangle + b) - 1] \quad (\text{II.8})$$

où les multiplicateurs de Lagrange notés  $\alpha_i$  sont tous positifs ou nuls. La résolution de ce problème d'optimisation consiste à trouver un point selle solution du problème de Lagrange. Écrivons les conditions de Karush-Kuhn-Tucker associées à ce problème d'optimisation. La recherche d'un extremum nécessite que les conditions suivantes soient vérifiées :

$$\begin{aligned} \frac{\partial L(\omega, b, \alpha)}{\partial b} &= 0 = \sum_{i=1}^n \alpha_i c_i \\ \frac{\partial L(\omega, b, \alpha)}{\partial \omega} &= 0 = \omega - \sum_{i=1}^n \alpha_i c_i X_i \end{aligned} \quad (\text{II.9})$$

La deuxième condition est très importante pour notre problème. Elle nous indique que l'hyperplan de séparation peut être exprimé uniquement à partir des réalisations dont le multiplicateur de Lagrange est non nul. De tels échantillons sont appelés les Vecteurs Supports. Ces Vecteurs Supports ont la particularité de se situer sur

la marge définie précédemment. Ceci se démontre par la condition d'optimalité de Karush-Kuhn-Tucker suivante :

$$\alpha_i (c_i (\langle \omega, X_i \rangle + b) - 1) = 0, \quad i = 1, \dots, n \quad (\text{II.10})$$

Les  $\alpha_i$  non nuls correspondent aux limitations actives du problème d'optimisation (ou contraintes actives de l'optimisation). Ces contraintes actives ne sont autres que les points sur la marge. Cela conduit à la solution suivante :

$$\omega = \sum_{j=1}^S \alpha_j c_j X_j \quad (\text{II.11})$$

où  $S$  est le nombre de vecteurs supports. Il est clairement montré que l'hyperplan est uniquement défini par les vecteurs supports, *ie.* les points les plus proches, ce qui est pressenti intuitivement : pour définir la frontière entre les domaines il suffit de s'appuyer sur les points proches de cette frontière. Cette propriété est essentielle pour la suite : **pour affiner la séparation, seules les réalisations dans la marge sont utiles, les autres peuvent être classées sans calcul supplémentaire à partir de la connaissance du séparateur SVM.**

### II.2.2 Le cas non linéaire - méthode à noyaux

#### Position du problème

Le principe est le suivant : en utilisant un projecteur non linéaire  $\Phi$ , une transformation de l'espace de départ est réalisée. Cette transformation effectue une projection de l'espace de départ dans l'espace dit support, espace qui présente la particularité d'avoir une dimension supérieure à l'espace de départ. Le lecteur trouvera quelques données sur ce qu'on appelle ici le *Kernel Trick*, en annexe VII. La figure II.2 illustre cette transformation de l'espace. L'espace support permet de retrouver le cas précédent des classes linéairement séparables, la formulation du séparateur reste la même, mais le produit scalaire (fonction noyau linéaire) est remplacé par une fonction noyau  $K$  plus générale.

Cela se traduit dans les équations de la manière suivante :

– rappelons la fonction de classification linéaire II.2, elle s'écrit comme suit :

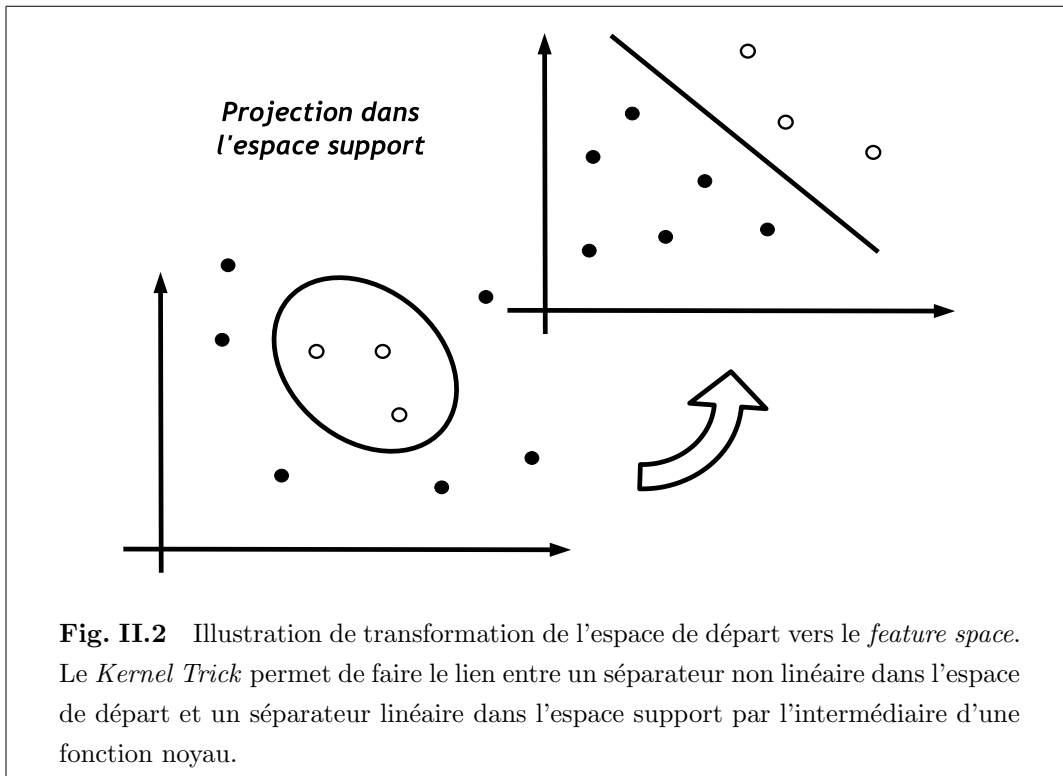
$$c(X) = \text{sgn}(G(X)) = \text{sgn} \left( \sum_{i=1}^n \alpha_i c_i (\langle X_i, X \rangle + b) \right) \quad (\text{II.12})$$

- en remplaçant le produit scalaire par une fonction noyau, on obtient l'expression suivante :

$$c(X) = \text{sgn}(G(X)) = \text{sgn} \left( \sum_{i=1}^n \alpha_i c_i (K(X_i, X) + b) \right) \quad (\text{II.13})$$

où :  $K(X_1, X_2) = \langle \Phi(X_1), \Phi(X_2) \rangle$

$\Phi()$  est la fonction de projection. Cette fonction n'a pas besoin d'être explicitée, nous utilisons dans la pratique uniquement une fonction noyau  $K$ . La construction d'une fonction noyau permet de travailler dans un espace de Hilbert de dimension infinie. Cette dimension permet d'obtenir une classification correcte en utilisant toujours un hyperplan.



Citons quelques fonctions noyau couramment utilisées :

$$\begin{aligned}
 \text{Radial Basis Function (RBF)} : K &= \exp \left( -\frac{1}{2\sigma^2} \|X_1 - X_2\|^2 \right) \\
 \text{Polynomiale} : K &= (\langle X_1, X_2 \rangle + c)^d \\
 \text{Fourier} : K &= \frac{(m + \frac{1}{2})(X_1 - X_2)}{2 \sin \left( \frac{X_1 - X_2}{2} \right)} \\
 \text{Sigmoïdale} : K &= \tanh (-\kappa \langle X_1 - X_2 \rangle + \theta)
 \end{aligned} \tag{II.14}$$

Selon la fonction choisie, un ou plusieurs paramètres sont à ajuster par rapport au problème étudié. Quelques informations concernant les choix des paramètres dans le cadre d'une utilisation basique des SVM sont disponibles dans un guide pratique pour la classification : [HCL98]. Nous présentons (et utilisons) quelques pistes dans le cadre de l'utilisation des SVM pour la fiabilité, cf. chapitre III.

### La résolution du problème d'optimisation

Dans le cas de figure non linéaire, il est préférable de résoudre le problème d'optimisation Lagrangienne sous sa formulation duale. Le problème devient une maximisation :

$$\begin{aligned}
 \text{maximiser } \omega(\alpha) &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i c_i K(X_i, X_j) c_j \alpha_j \\
 \text{sous les contraintes} : \alpha_i &\geq 0 \text{ et } \sum_{i=1}^n \alpha_i c_i = 0
 \end{aligned} \tag{II.15}$$

Les  $\alpha_i$  solutions de ce problème sont ensuite substituées dans l'expression II.11 pour obtenir les valeurs des paramètres de l'hyperplan.

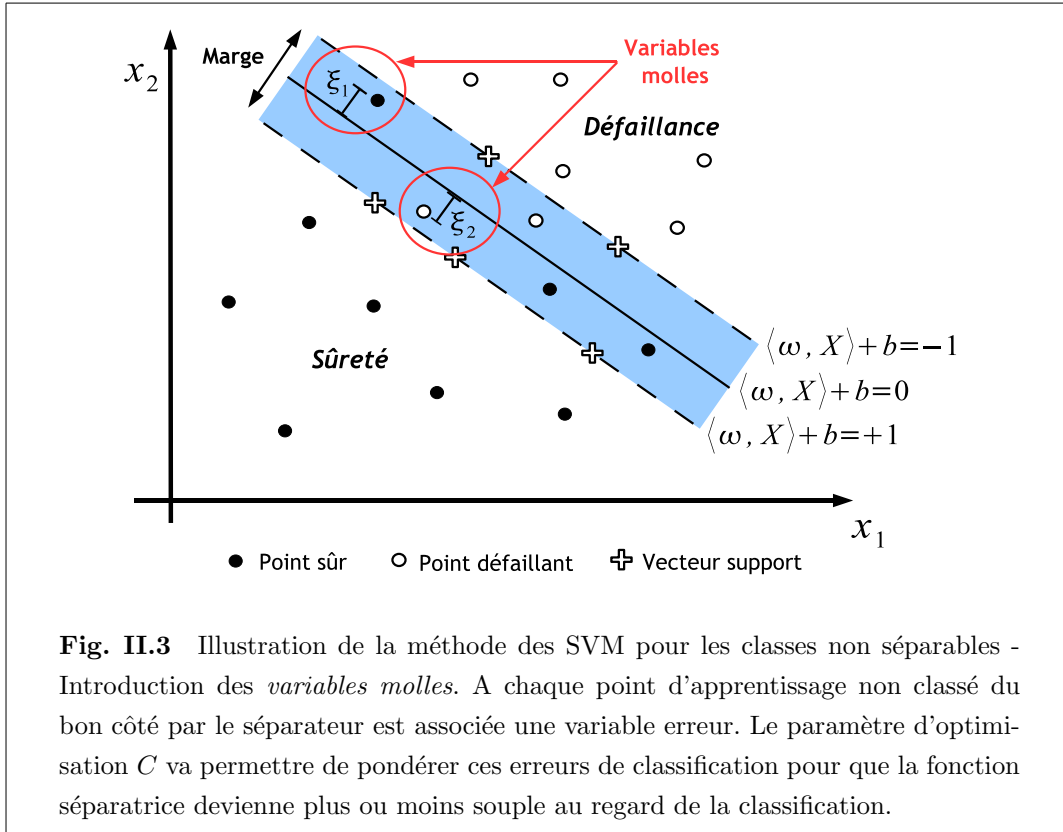
### II.2.3 Les classes non séparables

#### Position du problème

La classification des données en fiabilité peut être parfaite puisque les données sont numériques, un séparateur peut toujours être trouvé entre les 2 classes, même si celui-ci peut-être de forme complexe. Dans le cas de données expérimentales bruitées, la frontière peut être plus floue. Il est alors nécessaire de travailler avec la méthode décrite dans cette section, qui est une généralisation du cas des classes séparables présenté précédemment. La classification de données appartenant à des classes non



séparables est basée sur un concept de *marge molle*. Des variables nommées *slack variables*, ou *variables molles*, sont introduites. Ces variables  $\xi_i \geq 0, i = 1, \dots, n$  sont associées à chacune des  $n$  réalisations. Elles sont nulles en cas de bonne classification : la réalisation est placée du “bon côté” du séparateur. Dans le cas contraire,  $\xi_i$  correspond à la distance entre le séparateur et la réalisation  $x_i$ . Deux exemples d’écarts à la classification sont présentés sur la figure II.3.



Les contraintes de l'optimisation deviennent :

$$c_i (\langle \omega, X_i \rangle + b) \geq 1 - \xi_i \quad (\text{II.16})$$

ainsi le Lagrangien du problème d'optimisation s'écrit :

$$\min \frac{\|\omega\|^2}{2} + C \sum_{i=1}^n \xi_i \quad (\text{II.17})$$

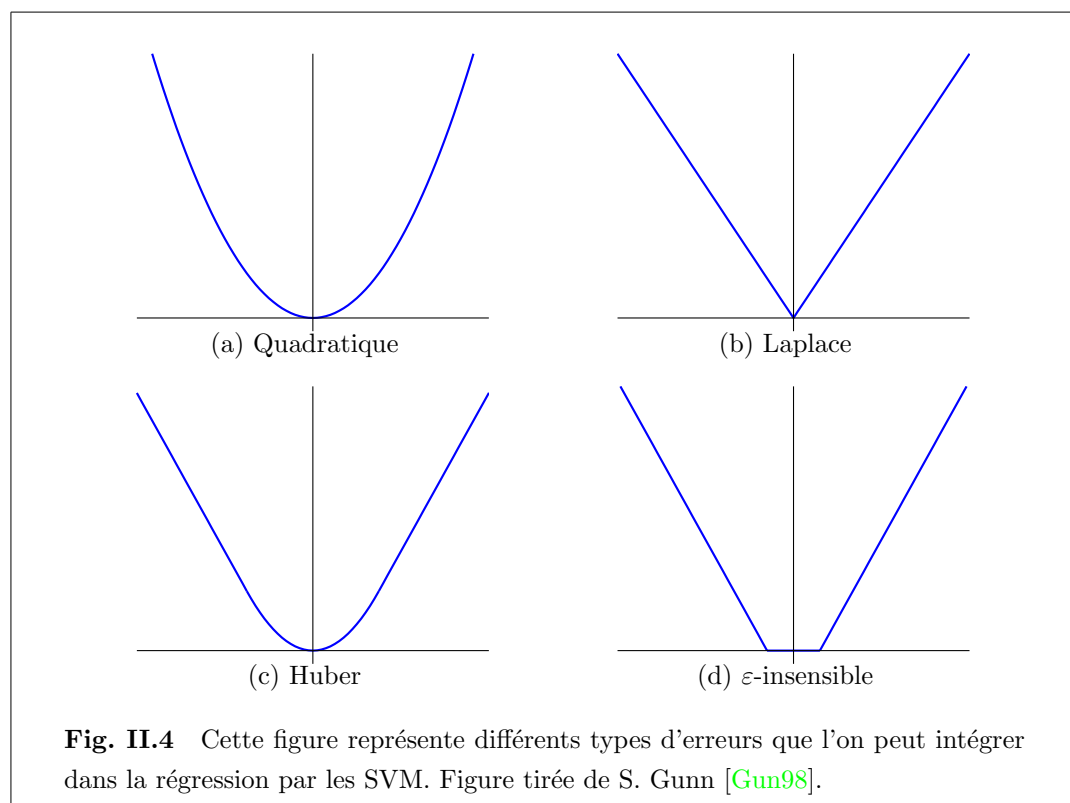
sous la contrainte :  $c_i (\langle \omega, X \rangle + b) \geq 1 - \xi_i, i = 1, \dots, n$

### Résolution de l'optimisation

Nous cherchons une solution  $(\omega, b)$  pour une constante  $C > 0$  telle que la marge soit maximale et le nombre d'erreurs de classification soit minimal. Le terme  $\sum_{i=1}^n \xi_i$  est une forme simplifiée de la somme des erreurs de classification. La fonction optimale solution du problème dans le *feature space* est appelée : *soft-margin hyperplane*. La constante  $C$  est fixée selon le choix de l'utilisateur : la valeur choisie permet de trouver un compromis entre erreur de classification et complexité de la fonction solution.

## II.3 Régression par SVM

### Position du problème



Les Machines à Vecteurs Supports (SVM) sont applicables **en classification** comme **en régression**, même s'il est clairement plus simple d'associer une classe à une réalisation que d'estimer la dépendance numérique des variables. Les résultats de la régression par les SVM sont encourageants si l'on se réfère à [CGS03], qui

compare différentes méthodes de régression. C'est sur la base de la méthode pour les classes non séparables que se généralise la méthode de régression par les SVM. Une présentation complète est disponible dans [SS03b] : “*tutorial on regression*”. Le concept de *soft margin* est remplacé par celui de  $\varepsilon$ -insensitive loss function qui est défini comme suit :

$$|y - f(x)|_\varepsilon = \max(0, |y - f(x)| - \varepsilon) \quad (\text{II.18})$$

où  $\varepsilon$  est un paramètre de précision de la régression,  $f(x)$  la fonction de régression recherchée et  $y$  la valeur prise en  $x_i$ . Cette fonction erreur peut prendre une forme classique comme celle de la fonction quadratique, cf. figure II.4. La plus usitée en régression SVM reste cependant la fonction  $\varepsilon$ -insensitive.

Ensuite, on introduit les *slack variables*  $\xi_i$  et  $\xi_i^*$ , variables similaires à celles utilisées dans le cadre de la classification *soft-margin*. Il s'agit alors de résoudre le problème d'optimisation suivant :

$$\begin{aligned} & \text{Minimiser } \frac{\|\omega\|^2}{2} + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ & \text{sous les contraintes : } \begin{cases} y_i - \langle \omega, X_i \rangle + b \leq \varepsilon + \xi_i \\ \langle \omega, X_i \rangle - b - y_i \leq \varepsilon + \xi_i^*, \quad i = 1, \dots, n \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (\text{II.19})$$

où la constante  $C$  joue le même rôle que dans le cas précédent : c'est un paramètre de contrôle entre complexité et erreur tolérée au delà de la précision  $\varepsilon$ . C'est ce compromis qui permet aux SVM de contrôler un bruit éventuel présent dans les réalisations, et/ou la non linéarité de l'état-limite. Cet ajustement apporte une souplesse à la méthode par rapport à la forme de la fonction à approcher par régression.

### La résolution du problème d'optimisation

La résolution du problème de régression est là encore effectuée en faisant appel au Lagrangien :

$$\begin{aligned} L(b, \omega, \alpha) = & \frac{\|\omega\|^2}{2} + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ & - \sum_{i=1}^n \alpha_i [\varepsilon + \xi_i - y_i + \langle \omega, X_i \rangle - b] \\ & - \sum_{i=1}^n \alpha_i^* [\varepsilon + \xi_i^* + y_i - \langle \omega, X_i \rangle + b] - \sum_{i=1}^n (\eta_i \xi_i + \eta_i^* \xi_i^*) \end{aligned} \quad (\text{II.20})$$

où les coefficients de Lagrange  $\alpha_i$ ,  $\alpha_i^*$ ,  $\eta_i$  et  $\eta_i^*$  sont positifs ou nuls. L'optimisation aboutit à une maximisation du Lagrangien par rapport aux variables ( $\alpha_i$ ,  $\alpha_i^*$ ,  $\eta_i$  et  $\eta_i^*$ ) et une minimisation par rapport aux variables ( $\omega$ ,  $b$ ,  $\xi$  et  $\xi^*$ ). L'expression de la fonction de régression est la suivante :

$$f(X) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \langle X, X_i \rangle - b \quad (\text{II.21})$$

Dans le cas général où le modèle de régression n'est pas linéaire, l'utilisation d'une fonction noyau est bien entendu possible et l'expression suivante est alors obtenue :

$$f(X) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(X_i, X) - b \quad (\text{II.22})$$

L'intérêt de cette méthode est présenté dans la section suivante II.4 en comparaison avec les réseaux de neurones.

## II.4 SVM ou réseaux de neurones

La structure des SVM s'apparente aux réseaux de neurones et est issue de la même théorie, ou du moins du même domaine : l'apprentissage statistique. Il est donc normal de se poser la question des différences entre ces deux outils. En effet, les réseaux de neurones ont montré quelques faiblesses pour une utilisation en fiabilité, notamment en raison du surapprentissage mal maîtrisé qui nécessite le calcul d'un certain nombre de points de validation.

Notons plusieurs différences, qui sont en fait des avantages pour les SVM par rapport aux réseaux de neurones :

- Les SVM disposent du concept de marge, concept qui possède une interprétation mathématique simple, et permet d'implémenter un algorithme de classification de manière incrémentale. La zone de marge correspond en fait à une zone d'incertitude. Pour améliorer ou affiner la marge, de nouveaux points d'apprentissage sont à choisir dans la marge.
- Les SVM appliquent le *principe SRM* (*Structured Risk Minimization*) (cf. II.4.2).
- L'optimisation aboutissant au calcul des paramètres de la fonction mathématique recherchée (la maximisation de la marge) ne nécessite qu'une optimisation quadratique. L'optimisation aboutit à la minimisation d'une fonction quadratique en  $\alpha$ . L'optimum atteint est alors global et il est trouvé par une méthode

d'optimisation de type SQP :

$$\begin{aligned} \text{Maximiser } \omega(\alpha) &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i c_i \langle X_i, X_j \rangle c_j \alpha_j \\ \text{sous les contraintes : } &\alpha_i \geq 0 \text{ et } \sum_{i=1}^n \alpha_i c_i = 0 \end{aligned} \quad (\text{II.23})$$

- La recherche des paramètres et la *VC dimension* (Vapnik-Chervonenkis) qui contrôle la complexité du modèle et la généralisation de celui-ci, permettent de passer outre le problème de la dimension. Ainsi **le nombre d'échantillons et la capacité de généralisation sont indépendants de la dimension du problème.**

Plus qu'une comparaison entre réseaux de neurones et SVM, cette section présente les concepts fondamentaux de l'apprentissage statistique. Ces concepts sont nécessaires à la compréhension des apports des SVM et de leurs capacités de généralisation. Les parties suivantes rappellent ces quelques points importants.

#### II.4.1 ERM et principe de régularisation

ERM ou *Empirical Risk Minimization*.

##### Fonctions coûts

Il est utile de définir au préalable quelques fonctions permettant de déterminer le degré d'apprentissage :

- tout d'abord la fonction de coût (ou de perte)  $L(y, h(X, \omega))$  de la machine d'apprentissage.  $y$  est la fonction à approximer et  $h(X, \omega)$  la machine d'apprentissage, où  $\omega$  est un vecteur de paramètres définissant la machine d'apprentissage.
- On définit également une fonction risque qui est la valeur espérée de la fonction coût.

$$R(\omega) = \int L(y, h(X, \omega)) dP_{X,y}(X, y) \quad (\text{II.24})$$

Si la distribution  $P_{X,y}(X, y)$  est dérivable, l'expression de l'intégrale est réduite :

$$R(\omega) = \int L(y, h(X, \omega)) p_{X,y}(X, y) dX dy \quad (\text{II.25})$$

Comme la densité conjointe  $p$  est inconnue, le risque est souvent approximé par le risque empirique  $R_e(\omega)$  évalué à partir d'un échantillon de  $n$  individus calculés jusque là. Il s'agit alors de minimiser un tel risque. Cette approximation est connue sous le

nom du *principe de minimisation du risque empirique, principe ERM (Empirical Risk Minimization)*.

### Principe ERM

Ce concept peut se traduire dans les différentes disciplines de l'apprentissage.

**(a) Estimation de densité.** Dans ce cadre, la fonction coût est la suivante :

$$L(h(X, \omega)) = -\ln(h(X, \omega)) \quad (\text{II.26})$$

d'après le principe de maximum de vraisemblance. Le risque devient :

$$R(\omega) = -\int \ln(h(X, \omega)) p_X(X) dX \quad (\text{II.27})$$

Le risque empirique qui en découle est le suivant :

$$R_e(\omega, n) = \frac{1}{n} \sum_{i=1}^n L(h(X_i, \omega)) \quad (\text{II.28})$$

**(b) Régression.** Le modèle de régression est la fonction  $g(X)$  telle que dans l'égalité  $y = g(X) + \epsilon$ , le carré de l'erreur  $\epsilon$  soit minimal. Pour résoudre ce problème, nous avons une famille de fonctions  $h(X, \omega)$  qui contiennent ou non la fonction  $g(X)$ . Nous cherchons alors les paramètres  $\omega$  qui minimisent la fonction coût :

$$L(y, h(X, \omega)) = (y - h(X, \omega))^2 \quad (\text{II.29})$$

Le risque associé à cette fonction  $L$  :

$$R(\omega) = \int (y - h(X, \omega))^2 p_{X,y}(X, y) dX dy \quad (\text{II.30})$$

Dans le cas d'une erreur  $\epsilon$  gaussienne, ce risque empirique s'écrit comme suit :

$$R_e(\omega, n) = \frac{1}{n} \sum_{i=1}^n (y_i - h(X_i, \omega))^2 \quad (\text{II.31})$$

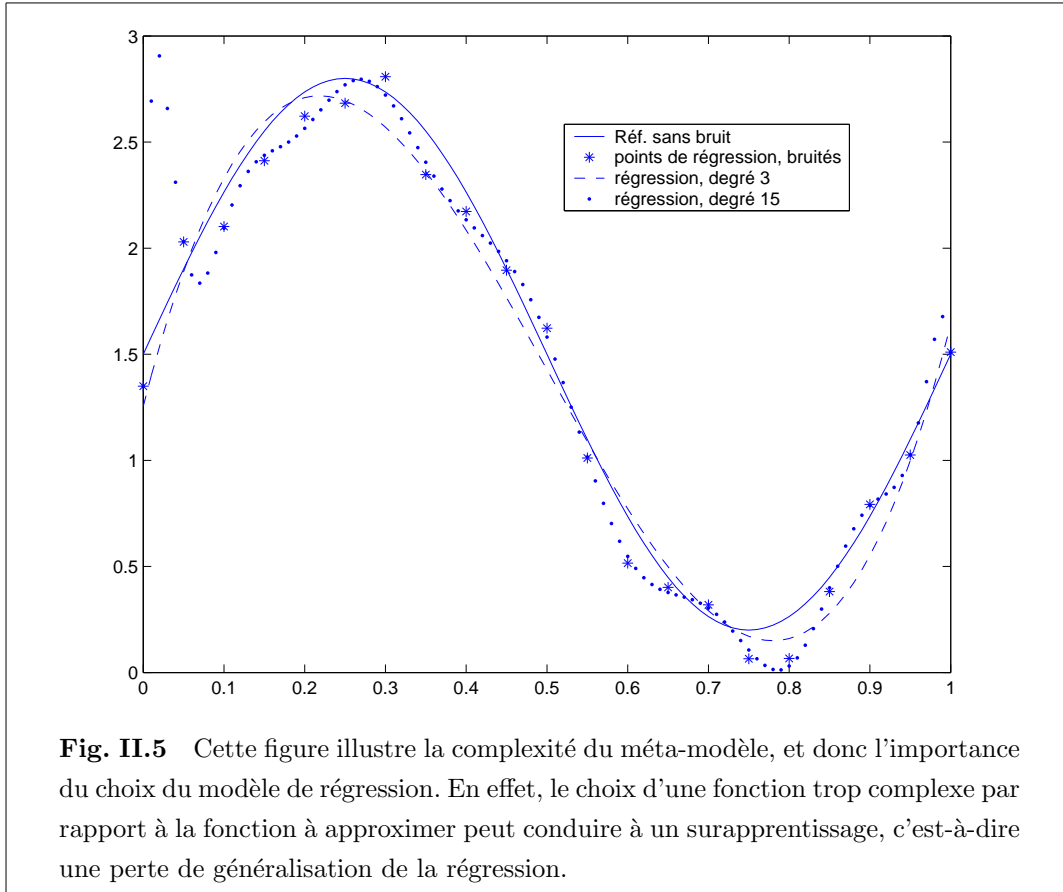
**(c) Classification.** Soit  $c(X) = \{+1; -1\}$  une fonction associant la valeur  $+1$  ou  $-1$  selon la classe de  $X$ . La fonction  $h(X, \omega) = \text{sgn}(g(X, \omega))$  est recherchée. L'objectif est de trouver la valeur de  $\omega$  qui minimise l'erreur calculée sur l'échantillon. La fonction coût est la suivante :

$$L(c, h(X, \omega)) = \begin{cases} 0, & \text{si } c = h(X, \omega) \\ 1, & \text{si } c \neq h(X, \omega) \end{cases} \quad (\text{II.32})$$

Le risque empirique associé est de la forme :

$$R_e(\omega, n) = \frac{1}{n} \sum_{i=1}^n L(c(X_i), h(X_i, \omega)) \quad (\text{II.33})$$

### Le surapprentissage



Dans le cadre de la régression par les moindres carrés, l'objectif est le calcul des coefficients du méta-modèle (modèle de régression) par la minimisation de ce risque empirique : c'est le principe ERM (*Empirical Risk Minimization*). Une régression qui suit ce principe, aussi bien faite soit-elle, est établie sur la base d'un modèle fixé à priori. Le problème est alors contenu dans le choix du méta-modèle. Ce choix est extrêmement important et doit être contrôlé puisqu'une fonction dont la complexité est excessive peut conduire à un *surapprentissage*.

Illustrons cette dernière idée par un exemple simple : la figure II.5 montre deux régressions par un polynôme de degré 3 et un polynôme de degré 15, sur des points

d'apprentissage issus de la fonction  $y(x) = 1,5 + 1,3\sin(2\pi x)$ , à laquelle a été ajouté un bruit gaussien. Le polynôme de degré 3 est globalement proche des points de régression, tandis que le polynôme de degré 15 passe par tous les points d'apprentissage. Ces 2 tracés de régression illustrent les notions d'erreur et de complexité. Le polynôme de degré 3 possède une erreur empirique plus importante que le polynôme de degré 15 mais ce dernier possède une erreur de généralisation plus importante que le polynôme de degré 3. Ceci montre que le modèle idéal doit venir d'un compromis entre erreur sur les échantillons, erreur empirique, et erreur de généralisation. C'est avec cet objectif que l'apprentissage des réseaux de neurones utilise des points de validation, sur lesquels l'erreur de généralisation est mesurée. L'erreur de généralisation augmente avec la complexité du modèle, ce qui est illustré par le polynôme de degré 15. Un modèle trop complexe va aboutir à un problème de surapprentissage. Ainsi, la minimisation du risque global va devoir être pondérée par le contrôle de la complexité du modèle de régression.

### Principe de régularisation

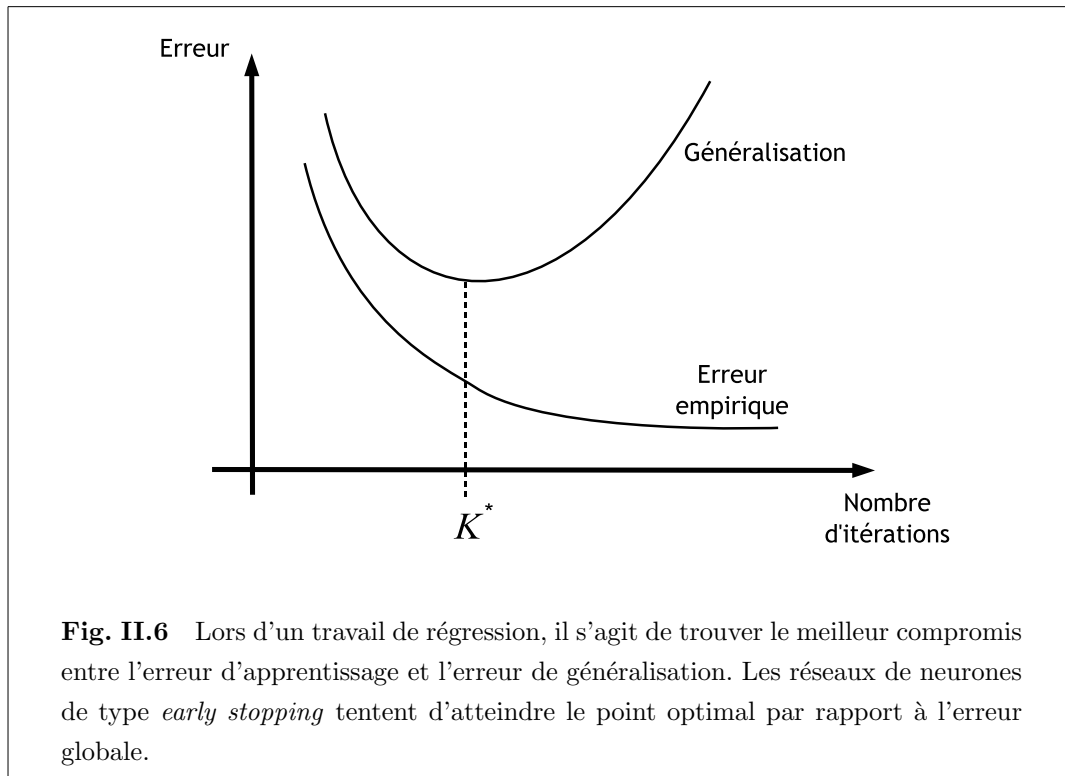
De l'idée de contrôle de la complexité du modèle est né le *principe de régularisation* (ou de pénalisation). Il s'agit ici, non plus de minimiser uniquement le risque empirique, mais d'y ajouter un terme qui va dépendre de la complexité de la fonction. Le problème d'optimisation s'écrit alors :

$$\text{Minimiser } R_r(\omega, n) = R_e(\omega, n) + \lambda \Psi(h(x, \omega)) \quad (\text{II.34})$$

où  $\Psi(h(x, \omega))$  est une fonction de régularisation qui dépend de la complexité du méta-modèle. Le coefficient  $\lambda$ , coefficient de Lagrange positif et non nul, permet de contrôler le poids de la complexité de la fonction sur le risque global. Ce principe est tout à fait intéressant et doit être pris en compte lors de régressions.

Dans le cadre des réseaux de neurones, la complexité du modèle est déterminée par le nombre de neurones sur la (ou les) couche(s) cachée(s). Le choix de ce paramètre va alors être testé par essais et erreurs (ce qui augmente le temps de traitement), et nécessite le calcul de plusieurs points de validation pour approximer l'erreur sur la généralisation de la fonction mathématique obtenue, l'objectif étant d'obtenir le meilleur compromis. Autre exemple, les réseaux de neurones MLP utilisent la technique dite de *early stopping*, ou arrêt anticipé : l'algorithme d'optimisation des paramètres du réseau est arrêté avant la convergence complète. Ceci est effectué dans le but de se trouver dans le cas illustré sur la figure II.6.





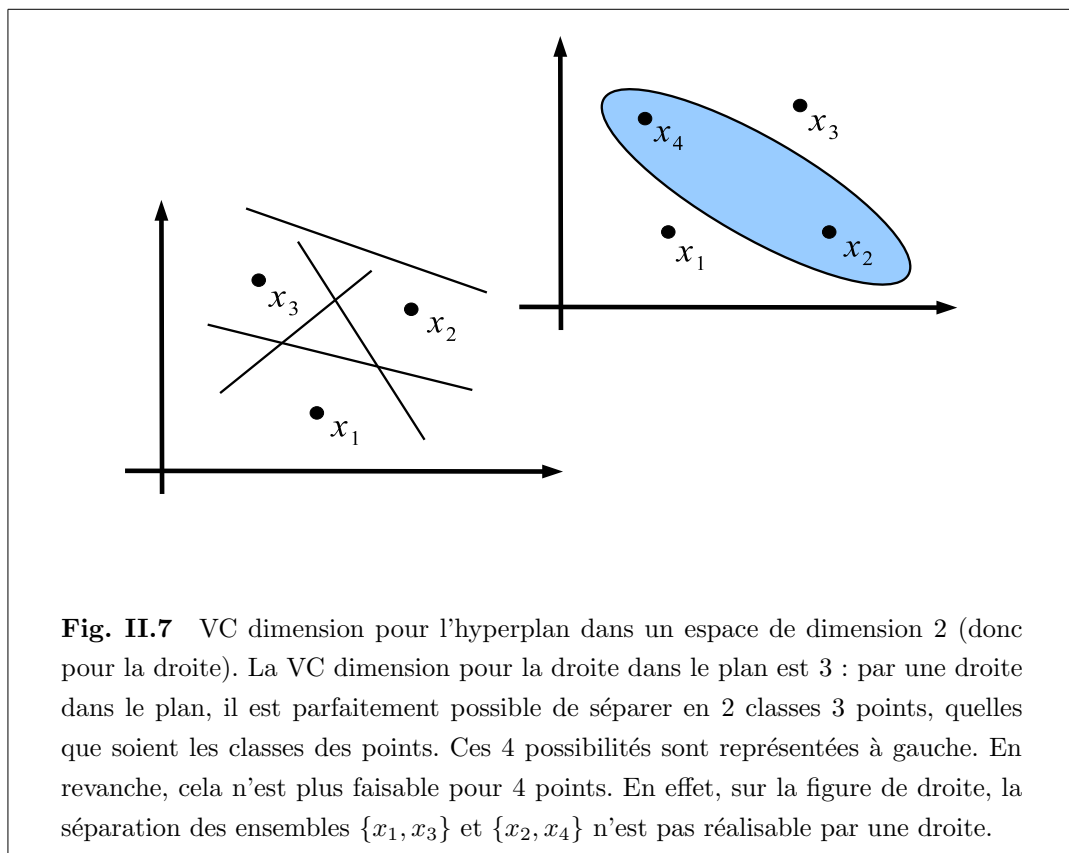
Dans le cadre de la régression classique, le méta-modèle est fixé *a priori*, et l'objectif est d'obtenir l'erreur empirique minimale à complexité fixée. Dans le cadre des SVM, un principe beaucoup plus évolué est appliqué de manière intrinsèque. Ce principe est nommé SRM, pour *Structured Risk Minimization*. Le but de ce principe est d'incorporer au processus d'apprentissage, une optimisation du méta-modèle.

## II.4.2 VC dimension et SRM

### VC dimension

Avant de présenter le principe de *Structured Risk Minimization*, il est indispensable d'aborder la notion de *Vapnik-Chervonenkis dimension* [VA71]. Simplement, la *VC dimension* d'une fonction correspond à une mesure de sa complexité. La définition est la suivante : c'est le nombre maximum  $v$  d'échantillons qu'il est possible de séparer par le modèle selon les  $2^v$  façons, où  $v = d + 1$  dans le cas d'un modèle linéaire, et  $d$  est la dimension de l'espace. Ceci est illustré dans le plan où  $v = 3$  pour la droite, comme le montre la figure II.7. Dans ce cas de figure, il y a  $2^v = 8$  façons de classer les 3 points :

- le groupe  $x_1, x_2, x_3$  en classe +1 et le reste de l'espace en classe -1 ;
- le groupe  $x_1, x_2$  en classe +1 et le groupe contenant  $x_3$  en classe -1 ;
- le groupe  $x_1$  en classe +1 et le groupe contenant  $x_2, x_3$  en classe -1 ;
- le groupe  $x_2$  en classe +1 et le groupe contenant  $x_1, x_3$  en classe -1 ;
- et leurs complémentaires, *i.e.* pour le premier cas par exemple,  $x_1, x_2, x_3$  passe en classe -1 et le reste de l'espace en classe +1. Ce qui fait bien 8 possibilités.



Sur la figure II.7, à gauche, toutes les possibilités de séparer 3 points sont réalisables par une droite, en revanche, il est impossible de séparer de toutes les façons 4 points : à droite de la figure II.7, les points  $x_2$  et  $x_4$  ne peuvent être séparés des autres réalisations par une droite. Il est important de retenir que cette dimension mesure la complexité d'un méta-modèle, avant de passer à l'explication du principe SRM.

### Bornes pour l'erreur de généralisation

Vapnik a démontré, dans une série de théorèmes [Vap98], l'existence de bornes pour le risque de généralisation, bornes liées à la dimension VC du méta-modèle. Pour la classification binaire en particulier, il a montré que pour une fonction  $Q(X_n, \omega)$  il y a une probabilité  $1 - \eta$  que le risque de classification admette l'inégalité suivante :

$$R(\omega) \leq R_e(\omega, n) + \frac{q}{2} \left( 1 + \sqrt{1 + \frac{4 + R_e(\omega, n)}{q}} \right) \quad (\text{II.35})$$

où :

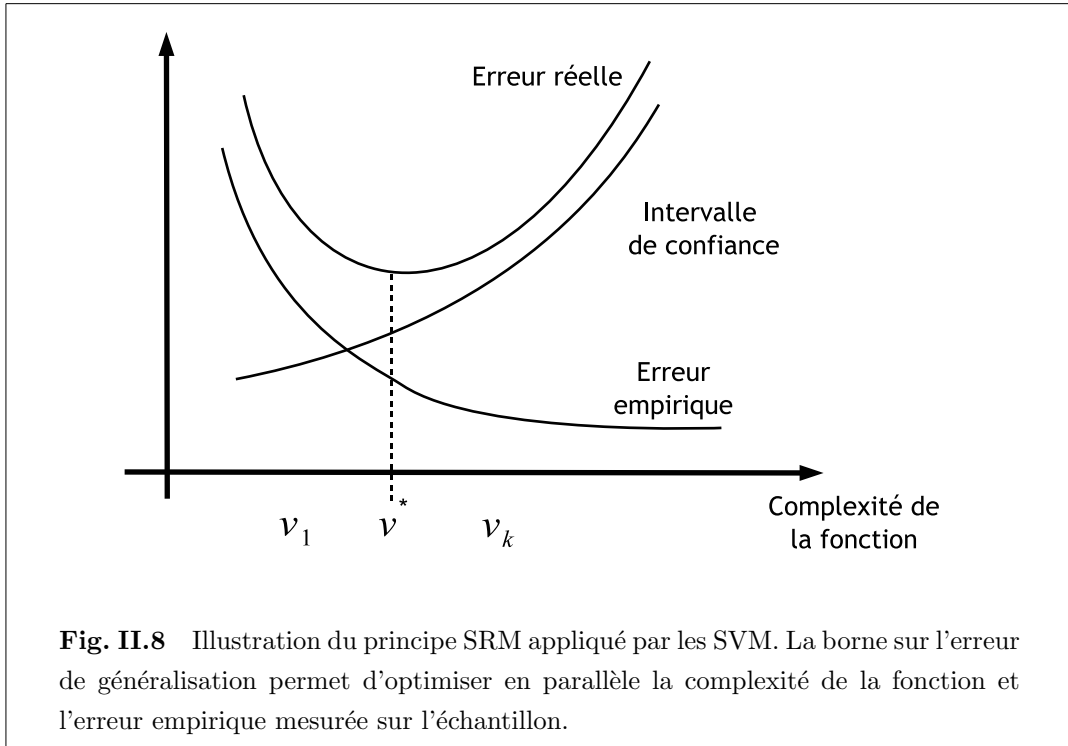
$$q = \frac{\alpha v (\ln(\beta n/v) + 1) - \ln(\eta/4)}{n} \quad (\text{II.36})$$

Les paramètres  $\alpha$  et  $\beta$  sont compris dans  $0 < \alpha \leq 4$  et  $0 < \beta \leq 2$ ,  $v$  indique la dimension VC du méta-modèle. Pour une VC dimension  $v$  donnée, si le rapport  $n/v$  est grand (nombre de points d'apprentissage sur dimension VC grand, ce qui est associé à un grand nombre de points), alors  $q$  est petit et il suffit simplement de minimiser le risque empirique, ce qui est fait habituellement. Dans le cas contraire, pour un petit nombre de points, il est nécessaire de minimiser les deux termes du risque pour minimiser la somme. Dans ce cas de figure, et lorsque les points d'apprentissage sont coûteux (comme c'est le cas en fiabilité), il est utile de contrôler la variable  $v$  pour minimiser le risque, *i.e.* contrôler la complexité du méta-modèle que l'on utilise.

### Structured Risk Minimization (SRM)

Ces bornes permettent donc de minimiser le risque global d'un séparateur à partir d'un échantillon constitué de quelques points, par le contrôle de la variable dimension VC.

Le but de l'optimisation est alors de trouver la valeur de VC qui va minimiser la somme du risque empirique et de l'intervalle de confiance sur le risque de généralisation, figure II.8. C'est le principe de minimisation du risque structuré (SRM). Il est possible d'obtenir, pour un minimum de points d'apprentissage, un séparateur avec une capacité maximale de généralisation (capacité maximale autorisée pour le nombre de points utilisé). Cette optimisation est réalisée de manière intrinsèque dans le concept de maximisation de la marge qui est à la base de la méthode des SVM.



### Le cas de la régression

Dans le cadre de la régression, les bornes du risque sont différentes :

$$R(\omega) \leq \frac{R_e(\omega, n)}{\max(0, 1 - \sqrt{q})} \quad (\text{II.37})$$

Cette expression montre que le choix du méta-modèle a un effet multiplicatif sur le risque alors qu'il n'avait qu'un effet additif sur le risque en classification. Ceci prouve que le problème de régression est plus complexe que le problème de classification, ce qui semble naturel. **L'estimation de la valeur d'une fonction est plus sensible au méta-modèle que l'estimation de son signe.** Une optimisation de la complexité du modèle devient alors indispensable.

### II.4.3 La grande dimension

Dans le cadre de la classification, l'effet de la dimension est quasiment inverse à celui attendu habituellement dans les disciplines mathématiques. En effet, plus la dimension de l'espace augmente, plus il est aisé de séparer les données par un simple hyperplan. La probabilité de "bonne classification" binaire de  $n$  données par

un hyperplan dans un espace de dimension  $d$  s'écrit, [Fin99] :

$$P(n, d) = \begin{cases} 1, & n \leq d + 1 \\ \left(\frac{1}{2}\right)^{n-1} \sum_{k=0}^d \binom{n-1}{k}, & n > d + 1 \end{cases} \quad (\text{II.38})$$

Par conséquent, si  $\frac{n}{d} < 2$ , il y a une très forte probabilité de bonne classification ; si  $d = n - 1$ , la bonne classification est garantie.

Cette propriété ne montre pas que la dimension de l'espace n'a pas d'influence pour toute utilisation des SVM, mais que la classification en tire un avantage certain.

## II.5 Optimisation des SVM

Un des problèmes de l'utilisation des SVM qui reste ouvert est lié au choix des paramètres des fonctions noyaux. Pour optimiser les résultats des classifications ou régressions en SVM, par un choix optimal des paramètres, différentes techniques sont possibles. La plus connue, et la plus utilisée, est celle de la validation croisée, (*cross validation*). D'autres utilisent des algorithmes d'optimisation pour améliorer les résultats par un choix optimal de paramètres.

La section suivante présente différentes techniques pour l'optimisation des différents paramètres des SVM.

### II.5.1 Le choix des paramètres

Plusieurs paramètres sont à fixer lors de la résolution du problème d'optimisation pour les SVM. Plusieurs méthodes sont disponibles pour optimiser la valeur de ces paramètres. Nous présentons ici l'approche la plus répandue : la validation croisée (issue des techniques de *bootstrapping* [ET93, Jul97]). Cette technique est également la plus simple à mettre en place.

Il existe à l'heure actuelle de nombreuses recherches dans le domaine de l'optimisation des paramètres : par exemple l'approche par gradients proposée par O.Chapelle [CVBM02, Cha02], et des méthodes plus avancées : [BG04a, ARRZ03].

Toutes ces techniques sont très utiles lorsque le problème traité est un problème de régression qui nécessite plus de précautions. Il faut cependant prendre garde au surapprentissage que peut entraîner une recherche optimale des paramètres par rapport à un échantillon de données. Les utilisateurs des SVM montrent qu'il y a moins d'influence du choix de la fonction noyau pour la classification, surtout pour des données séparables, ce qui est le cas en fiabilité.

Nous nous devons de présenter ces approches d'optimisation des paramètres, même si celles-ci sont utilisées dans les cas où l'on possède de très nombreuses données. Il s'agit dans notre situation d'exploiter les qualités de généralisation des SVM, plus que leur qualité de séparateur optimal dans le cadre de données complexes.

### Le paramètre $C$

En premier lieu, il convient de choisir la valeur de  $C$ . Cette valeur correspond au compromis entre mauvaise classification et complexité de la fonction, soit entre erreur empirique et erreur de généralisation. La valeur de ce paramètre est surtout très importante en régression, ou dans le cadre de la classification de classes non séparables.

### Le(s) paramètre(s) du noyau

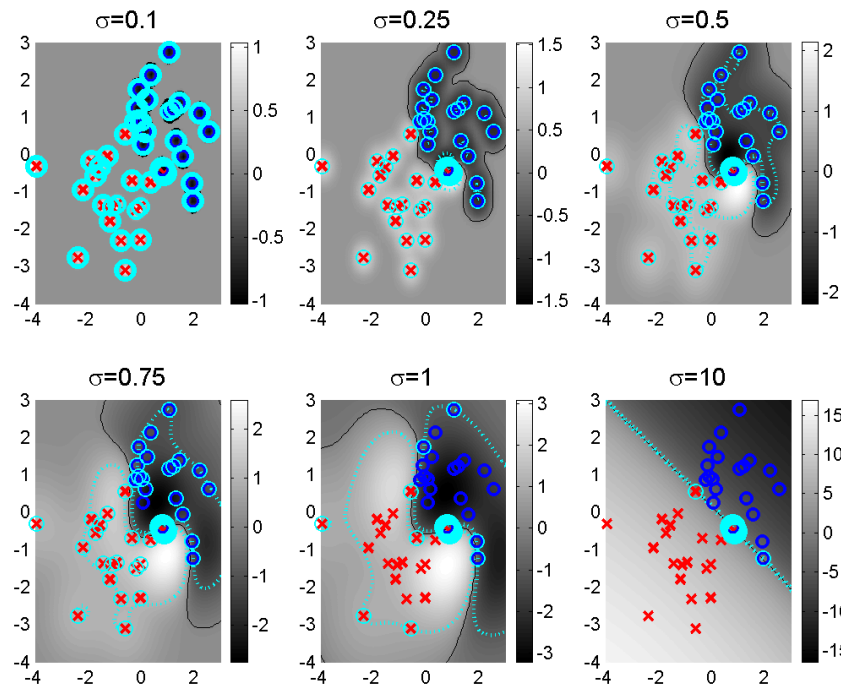
En second lieu, il est nécessaire de fixer les paramètres de la fonction noyau. Par exemple, pour le choix de la fonction noyau RBF, il est nécessaire de fixer une valeur à  $\sigma$ , cf. équation II.14. Sur un exemple de classification, la figure II.9 illustre l'influence de la valeur de ce paramètre.

## II.5.2 Validation croisée

La méthode la plus répandue pour l'optimisation des paramètres porte le nom de méthode de validation croisée (*cross validation*).

Le concept de validation croisée est assez simple, il s'agit de calculer plusieurs séparateurs avec des valeurs différentes pour les paramètres. L'erreur est mesurée par rapport à l'échantillon de points dont la classe est connue, et le meilleur séparateur est conservé. Ce concept simple est très utile, mais peut aboutir à un surapprentissage. En effet, le fait de chercher la meilleure fonction sur nos données, par rapport à l'erreur empirique, peut engendrer une défaillance de la propriété de généralisation de la fonction obtenue.

Dans le cas de données déséquilibrées, *i.e.* dont le nombre de données par classe n'est pas du même ordre pour les différentes classes, il est possible de pondérer le calcul de l'erreur pour forcer le séparateur à ne pas oublier de points dans la classe la moins nombreuse. Typiquement dans le cas de la fiabilité des structures, il y a peu de points dans la classe de défaillance. Il est alors possible de conserver le séparateur qui a tendance à maximiser le nombre de points dans la défaillance.



**Fig. II.9** Choix du paramètre  $\sigma$  pour les fonctions RBF. Lorsque  $\sigma$  décroît, le séparateur SVM devient de plus en plus précis et de moins en moins général.

### II.5.3 Boosting algorithm

La technique de *boosting*, dont l'algorithme le plus connu est l'AdaBoost (les notions principales sont décrites dans [Wik07a]) est une technique d'apprentissage assez proche de la théorie des SVM. Elle peut être utilisée avec les différentes catégories de séparateurs pour améliorer leur performance en terme de généralisation.

Il s'agit de générer successivement plusieurs séparateurs, en partant d'un séparateur à très forte capacité de généralisation, puis de les combiner. L'idée est de réaliser l'apprentissage du séparateur  $k+1$  en le forçant à se plier autour des points mal classés à l'itération  $k$ . Le séparateur final est une combinaison linéaire des différents séparateurs obtenus.

Détail de l'algorithme AdaBoost appliqué aux SVM, nommé AdaBoostSVM, issu de X.Li [LWS05] :

1. Données d'entrée :
  - la base d'apprentissage  $(X_1, y_1), \dots, (X_N, y_N)$  ;
  - le paramètre initial de la fonction noyau RBF  $\sigma_{init}$  ;

- une valeur minimale de ce paramètre  $\sigma_{min}$  ;
  - un pas pour ce paramètre  $\sigma_{step}$ .
2. Initialisation des poids associés à chaque point d'apprentissage dans la fonction de classification:  $w_i^1 = \frac{1}{N}, i = 1, \dots, N$ .
  3. Tant que ( $\sigma > \sigma_{min}$ )
    - (a) Optimisation d'un SVM (RBF)  $h_t$  sur la base d'apprentissage.
    - (b) Évaluer l'erreur d'apprentissage de  $h_t$  :  $\epsilon_t = \sum_{i=1}^N \omega_i^t, y_i \neq h_t(X_i)$ .
    - (c) Si  $\epsilon_t > 0,5$  alors  $\sigma = \sigma - \sigma_{step}$  et retourner à (a).
    - (d) Fixer un poids au séparateur  $h_t$  :  $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)$ .
    - (e) Mettre à jour les poids :  $w_i^{t+1} = \frac{w_i^t \exp(-\alpha_t y_i h_t(X_i))}{C_t}$ , où  $C_t$  est la constante associée à l'optimisation de la SVM, et  $\sum_{i=1}^N w_i^{t+1} = 1$ .
  4. Sortie :  $f(X) = \text{sgn}\left(\sum_{i=1}^T \alpha_i h_t(X)\right)$ .

Dans cet algorithme, une priorité est donnée à la capacité de généralisation des SVM. En effet, le travail est en premier lieu effectué avec un large  $\sigma$ , ce qui correspond à une fonction assez régulière, (cf. figure II.9). Cet algorithme favorise le travail avec des SVM dont la capacité de généralisation est plus forte que la précision.

## II.6 Les outils associés

Il est possible, et même utile, d'associer à l'utilisation des SVM ou des réseaux de neurones, des techniques de compression de données. Celles-ci permettent de transformer un échantillon de réalisations en un autre de taille inférieure, tout en conservant leurs caractéristiques principales. Cette technique est également utilisée dans les éléments finis stochastiques par l'intermédiaire de la transformation de Karhunen Loève [GS91].

Dans la même optique, nous présentons ici l'*analyse en composantes principales*, couplée avec une approche par noyau ; et la *quantification vectorielle*.

Dans le cadre d'une population gaussienne très large, un tirage aléatoire permettrait d'aboutir à un résultat équivalent. Cependant, dans le cas d'une distribution moins régulière (par exemple une distribution de points dans la marge), un tirage aléatoire nous ferait perdre de l'information : l'objectif est de conserver les caracté-



ristiques principales de la répartition des données. Dans ce cas, l'utilisation d'outils tels que ceux présentés ici, a un intérêt non négligeable.

### II.6.1 Analyse en Composantes Principales (PCA)

Une présentation de cette technique est détaillée par J.Fortuna dans [FC04]. Cette méthode de réduction de la dimension d'une population est assez répandue. C'est une méthode linéaire : les réalisations dans le nouvel espace réduit sont des combinaisons linéaires des premières. Soit  $X_i$ ,  $i = 1, \dots, n$  un échantillon de vecteurs dans  $\mathbb{R}^d$ . Nous supposons que celui-ci est centré, soit :  $\sum_{i=1}^n X_i = 0$ .

Nous cherchons à écrire les vecteurs de l'échantillon sous la forme suivante :

$$X = \sum_{j=1}^d v_j Z_j \quad (\text{II.39})$$

où les  $Z_j$  forment une base orthonormale de  $\mathbb{R}^d$ . Ce qui implique dans ce cas que les coefficients sont définis par :

$$v_j = \langle Z_j, X \rangle \quad (\text{II.40})$$

Cette écriture va nous permettre de tronquer la sommation après  $r < d$  termes en minimisant l'erreur. Pour réaliser cette troncature, on diagonalise la matrice des covariances  $C$  :

$$C = \frac{1}{n} \sum_{i=1}^n X_i X_i^\top \quad (\text{II.41})$$

La diagonalisation nous impose de résoudre le problème aux valeurs propres suivant :

$$\lambda_k Z_k = C Z_k \quad (\text{II.42})$$

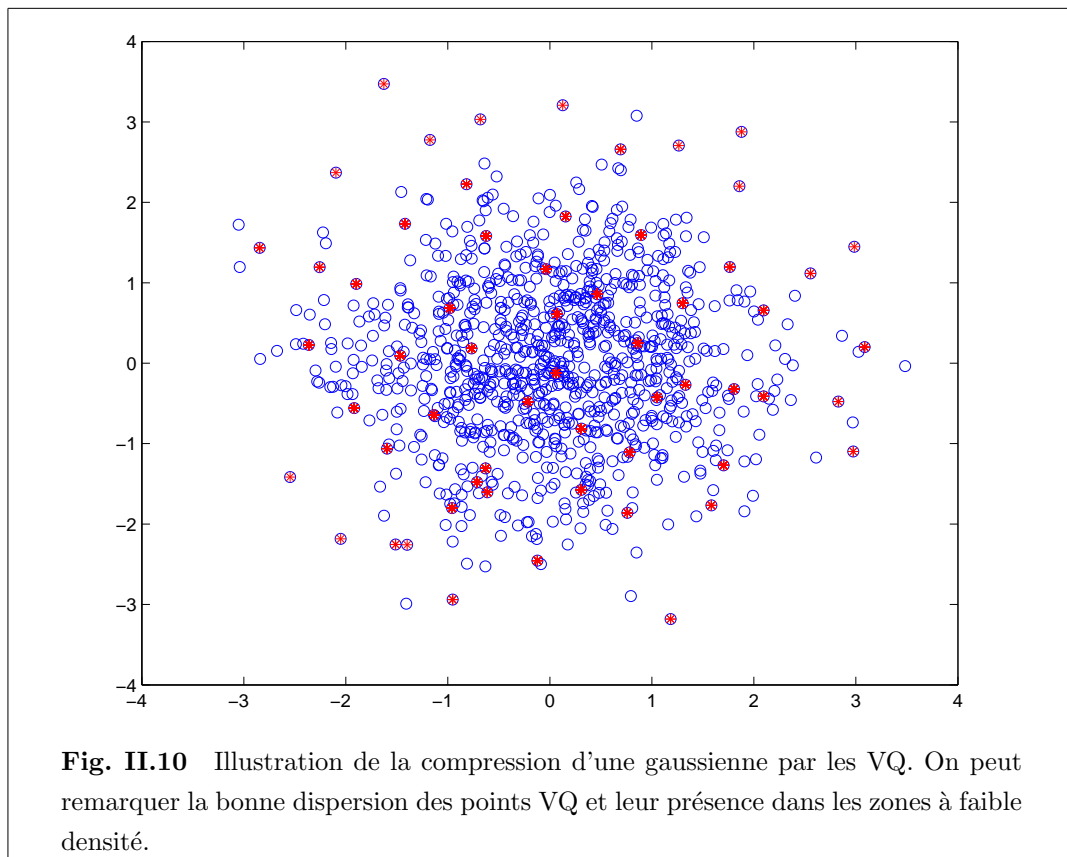
où les  $\lambda_k$  correspondent aux  $d$  valeurs propres et les  $Z_k$  aux vecteurs propres. Ensuite, les  $d$  valeurs propres solutions sont classées par ordre décroissant. Sélectionner les  $r < d$  premières valeurs propres consiste à conserver les variables qui expriment la plus large part de la variance de la population.

Une version non linéaire de l'analyse en composantes principales est également utilisée en apprentissage statistique. Cette extension utilise les fonctions noyau pour travailler dans un espace de Hilbert de très grande dimension. L'approche est basée sur une transformation non linéaire des données avant de résoudre le problème aux valeurs propres. Cette technique est donc souvent associée aux SVM, puisqu'elle possède des liens très forts avec les fonctions noyaux.

## II.6.2 Quantification Vectorielle (Vector Quantizer VQ)

La méthode de Quantification Vectorielle est un outil du traitement statistique de données et est utilisée dans de nombreuses applications informatiques telles que la compression d'image ou de vidéo. Cette technique est également employée dans le cadre de la classification et reconnaissance de forme. Cet outil permet de compresser un ensemble de données en conservant les caractéristiques principales de la distribution étudiée. L'ensemble des points candidats (points de la simulation appartenant à la marge) est découpé en régions de Voronoï, et dans chaque région est conservé le "barycentre", ou le point qui représente au mieux la distribution de la région. La figure II.10 permet de visualiser le résultat d'une compression par VQ.

Cette technique consiste à diminuer la taille d'un échantillon tout en conservant ses caractéristiques. C'est un moyen intéressant pour compresser une population générée pour des simulations de Monte-Carlo afin de diminuer le nombre d'appels à l'état-limite.



### Qu'est-ce qu'un VQ ?

Une quantification vectorielle  $Q$  transforme un ensemble de vecteurs de  $\mathbb{R}^d$  en un ensemble  $R = \{R_1, R_2, \dots, R_l\}$ , ensemble fini de vecteurs de  $\mathbb{R}^d$ . En fait, les  $l$  réalisations de  $R$  représentent les  $n$  réalisations  $X_i$ ,  $i = 1, \dots, n$ . Chaque  $R_i$  est un quantificateur, et à chaque  $R_i$  est liée une région de l'espace : la région de Voronoï, définie par :

$$V_i = \left\{ X \in \mathbb{R}^d \mid \|X - R_i\| \leq \|X - R_j\|, \forall i \neq j \right\} \quad (\text{II.43})$$

L'ensemble des régions de Voronoï forme une partition de l'espace des vecteurs :

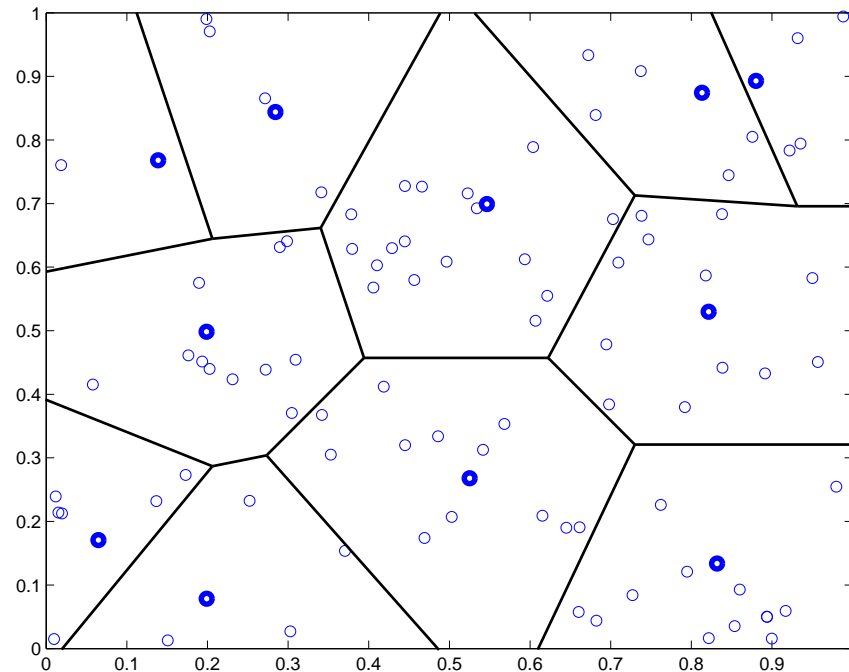
$$\begin{aligned} \bigcup_{i=1}^l V_i &= \mathbb{R}^d \\ \bigcap_{i=1}^l V_i &= \emptyset, \forall i \neq j \end{aligned} \quad (\text{II.44})$$

Chaque vecteur est représenté par le quantificateur associé à sa région de Voronoï. La figure II.11 présente la compression d'une distribution uniforme de points. A chaque quantificateur obtenu est associée une région de l'espace dont il est le représentant.

Le choix du quantificateur associé à un point dépend de la distance euclidienne entre points et quantificateur :

$$d(X, R_i) = \sqrt{\sum_{j=1}^l (x_j - r_{ij})^2} \quad (\text{II.45})$$

où  $x_j$  est la  $j$ -ième composante du vecteur  $X$  et  $r_{ij}$  est la  $j$ -ième composante du quantificateur  $R_i$ .



**Fig. II.11** À chaque région de Voronoï est associé un quantificateur, représentant des vecteurs de la région.

### Compression par VQ

Le VQ associe à un vecteur d'entrée  $X$ , l'index du quantificateur qui lui associe la plus faible distorsion. Dans notre cas, la distorsion correspond à la distance euclidienne.

Cette opération est réalisée à partir d'un *codebook* qui contient les quantificateurs.

Il reste à constituer ce *codebook*. Cette opération est réalisée par un algorithme proche du *K-Mean clustering*, cf. section II.6.3.

### II.6.3 Clustering

Les techniques de *clustering* permettent de réaliser des tâches dans la même idée que les VQ. Cette technique va permettre de compresser un échantillon de données. Cette théorie est largement utilisée en statistique. Elle permet de regrouper des données en groupes de mêmes caractéristiques. Dans notre cas, la caractéristique est la distance. L'algorithme *K-Mean clustering* est l'un des plus répandus. Un exemple est

présenté par S.Bandyopadhyay [BM02]. Il fonctionne de la manière suivante :

1. Choix du nombre de clusters  $k$  ;
2. Les données sont séparées en  $k$  clusters. Cette étape peut être réalisée de manière aléatoire ou comme suit :
  - les  $k$  premiers échantillons sont les  $k$  clusters ;
  - les autres points sont assignés au cluster qui en est le plus proche. Après chaque rapprochement, le centroïde est recalculé ;
3. Tous les échantillons sont pris ensuite dans l'ordre, et sont affectés au centroïde le plus proche. Chaque fois qu'un point change d'affectation, les 2 centroïdes concernés sont recalculés.
4. L'étape 3 est réitérée jusqu'à convergence.

La différence avec les VQ tient dans le fait qu'ici, les centroïdes n'appartiennent pas aux données de départ. Dans le cas des VQ, les points issus de l'algorithme sont des échantillons appartenant aux données à classer.

## II.7 Les outils utilisés

Le site [kernel-machines.org](http://kernel-machines.org) offre de nombreuses publications sur le thème des SVM. Il propose également plusieurs programmes, du plus éducatif au plus complet, permettant de mettre en œuvre l'outil SVM et d'en comprendre le principe. Parmi les outils disponibles, nous retenons SPIDER [SPI06], une boîte à outils pour MATLAB [MAT03], rassemblant les développements de nombreuses équipes de recherche et toujours en développement au Max Planck Institute, institut qui concentre de nombreux contributeurs importants dans le domaine de l'apprentissage statistique.

SPIDER permet l'utilisation des SVM pour la classification, la classification multi-classes, la régression, et propose également des algorithmes permettant d'optimiser les paramètres des vecteurs supports par comparaison de différents apprentissages, etc.

## II.8 Conclusion

Différents aspects de la mise en œuvre des SVM ont été présentés. La résolution de l'optimisation n'a pas été abordée. Il existe des algorithmes d'optimisation développés spécifiquement pour traiter des problèmes de SVM, qui sont caractérisés par une très

grande taille. En effet ces outils sont utilisés pour traiter de très grands ensembles de données. Pour les applications envisagées dans le cadre de cette thèse, la taille des échantillons reste très faible, c'est même un objectif. Les moyens permettant d'améliorer la résolution des problèmes d'optimisation associées à un grand nombre de données ne sont, pour cette raison, pas abordés puisqu'ils ne sont pas primordiaux pour les cas traités.

L'objectif de ces outils est de permettre la formulation mathématique d'un substitut de l'état-limite défini par le fiabiliste. L'idée est de créer la base de données de points d'apprentissage de manière optimale et d'optimiser une fonction SVM sur cette base. Les différents outils permettant ce travail ont été présentés dans ce chapitre. Il faut préciser que ce qui a été présenté vise à une classification, ou une régression, optimale par rapport à un ensemble de points donnés. Rien n'est dit sur la stratégie optimale d'ajout de points qui est un des objectifs des travaux de thèse.

Nous pouvons préciser quelques particularités des données issues d'un calcul numérique mécanique. Ci-dessous sont détaillées les propriétés qui caractérisent nos données et donc notre utilisation des SVM.

- *A priori*, les données sont parfaitement séparables, pas de marge molle.
- Les données sont choisies : nous sélectionnons les points de calcul du modèle.
- Les données sont adimensionnelles puisque nous travaillons après la transformation isoprobabiliste.
- L'espace à classer est défini par la zone de validité du modèle mécanique.



## Chapitre III

# SVM et Fiabilité

Ce chapitre détaille les avancées majeures réalisées dans notre travail. Sont présentés ici les algorithmes d'apprentissage construits spécifiquement pour l'application de méthodes fiabilistes.



## Sommaire

---

<b>III.1 Introduction</b>	<b>71</b>
<b>III.2 Fil conducteur</b>	<b>71</b>
III.2.1 La fonction d'état-limite	71
III.2.2 La représentation des SVM	73
<b>III.3 La méthode de J.E. Hurtado - SVM(H)</b>	<b>74</b>
III.3.1 Le principe de l'algorithme	74
III.3.2 Le détail pas à pas [Hur04a]	74
<b>III.4 Discussion et propositions d'améliorations de l'algorithme</b>	<b>77</b>
III.4.1 La fonction noyau	78
III.4.2 Le plan d'expériences initial - carré latin	79
III.4.3 Choix des points d'apprentissage dans la marge	83
III.4.4 Approche multi-échelle	85
III.4.5 Bilan : trois phases d'apprentissage	87
III.4.6 Critères de validation	88
<b>III.5 Bilan sur l'algorithme SMART</b>	<b>93</b>
III.5.1 Une vision globale	94
III.5.2 Une frontière explicite	95
III.5.3 Paramétrage de la méthode	95
III.5.4 En conclusion	95
<b>III.6 Subsets Simulations et Classification : méthode <sup>2</sup>SMART</b>	<b>96</b>
III.6.1 <i>Subsets simulations</i>	96
III.6.2 Algorithme SVM et <i>subsets simulations</i> - <sup>2</sup> SMART	104
<b>III.7 Conclusion</b>	<b>113</b>

---

## III.1 Introduction

L'OBJECTIF est de résoudre un problème de fiabilité en créant un substitut analytique de la fonction d'état-limite, en utilisant l'approche en classification entre les domaines de sûreté et de défaillance. La simulation de Monte-Carlo est ensuite abordée comme un problème de classification, de séparation des données simulées. Cette stratégie s'avère payante pour un problème de fiabilité grâce au concept de marge qui permet de créer une stratégie spécifique à l'apprentissage.

Dans un premier temps, le travail se concentre sur les simulations de Monte-Carlo qui sont, par ce biais, rendues très efficaces. Une critique constructive de l'algorithme d'apprentissage des SVM proposé par J.E. Hurtado [Hur04a] permet d'aboutir à un algorithme performant construit spécifiquement pour la fiabilité et l'obtention de la probabilité de défaillance.

Mais les limites de Monte-Carlo vont nous pousser, dans un second temps, à aborder avec le même point de vue d'autres types de simulations. Ce même principe est appliqué aux *subsets simulations* introduites par Au et Beck [AB01].

## III.2 Fil conducteur

### III.2.1 La fonction d'état-limite

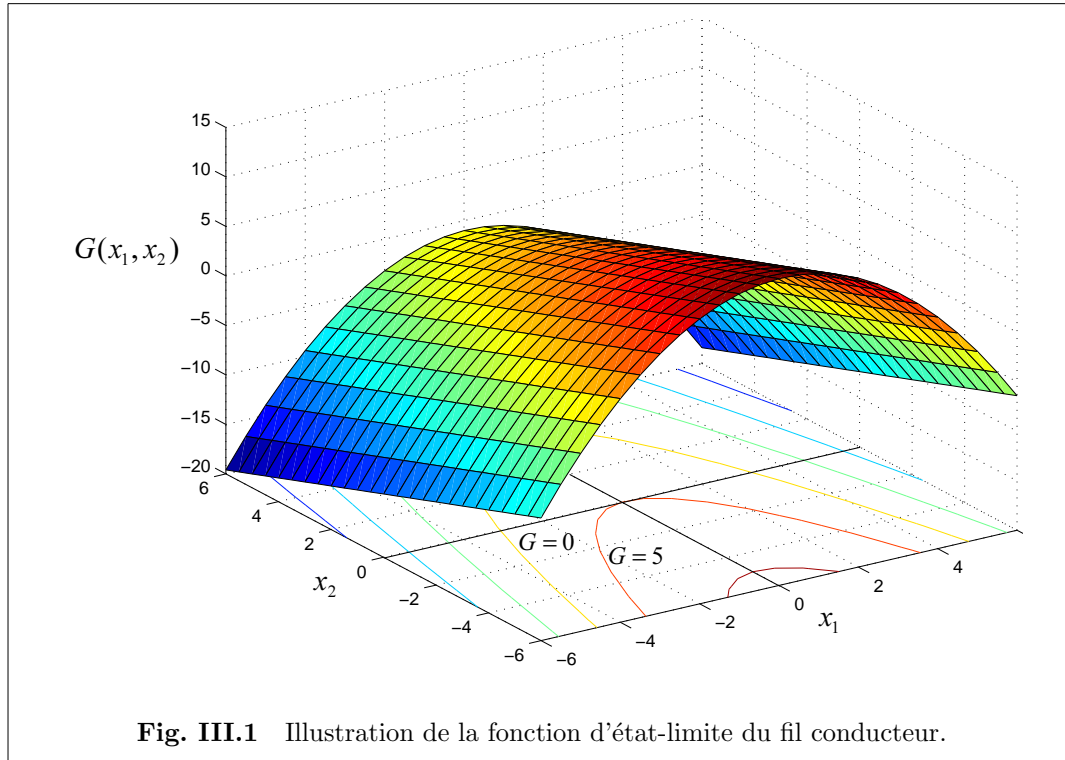
Pour l'illustration des différentes techniques présentées, un exemple simple [Ele05], cf. figure III.1, est utilisé comme fil conducteur de ce chapitre. Cette forme parabolique présente deux zones de maximum de densité, particularité qui en rend l'étude intéressante.

La fonction d'état-limite considérée est la suivante :

$$G(x_1, x_2) = \gamma - \frac{1}{2}(x_1 - 0, 1)^2 + x_2 \quad (\text{III.1})$$

Les variables  $x_1$  et  $x_2$  sont gaussiennes normées et centrées. La valeur de  $\gamma$  est fixée à 5 ou 10 pour faire varier la valeur de la probabilité de défaillance.

Dans le cadre de la fiabilité et de la classification, l'essentiel est de trouver une formulation analytique adéquate de la ligne de niveau  $G(x_1, x_2) = 0$ . Cet état-limite,

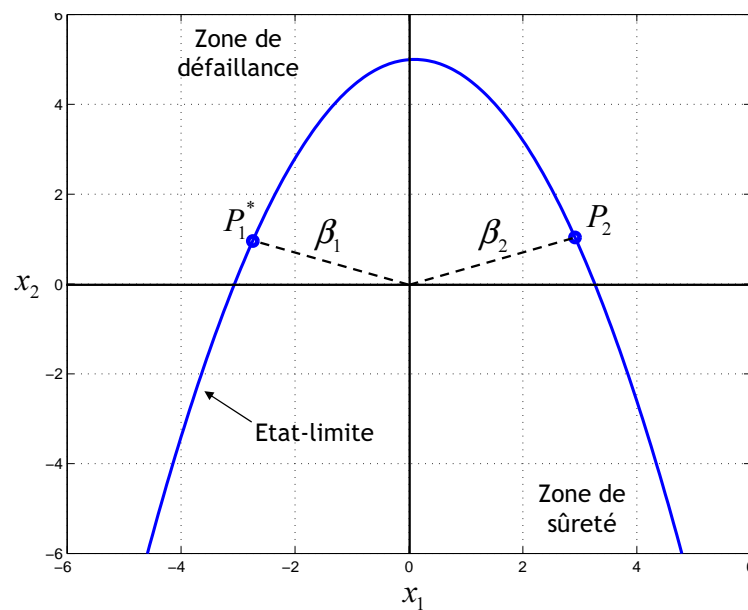


frontière entre domaine de défaillance et domaine de sûreté, est représenté sur la figure III.2.

Deux points de défaillance assez proches en terme d'indice de fiabilité existent et peuvent être calculés, cf. tableau III.1. L'intérêt de cet exemple est multiple. D'une part, il montre les limites de l'approximation de la probabilité de défaillance au premier ordre. En effet, l'approximation FORM ne permet pas d'obtenir la bonne valeur. Les résultats obtenus sont résumés dans le tableau III.1. D'autre part, cet exemple en 2 dimensions permet un affichage graphique de l'évolution de la fonction d'état-limite de substitution, au fur et à mesure de la progression de l'algorithme proposé dans ce chapitre.

	$P_f$	$\beta$	$x_1$	$x_2$
$P_1^*$	0,00183	2,90	-2,741	0,956
$P_2$	0,00093	3,11	2,916	1,086
Monte-Carlo ( $10^6$ simulations)	0,00306	2,74	—	—

**Tab. III.1** Résultats des simulations et analyse FORM sur l'état-limite.



**Fig. III.2** Frontière entre domaines de sûreté et de défaillance et positions des points de défaillance les plus probables.

### III.2.2 La représentation des SVM

Il convient ici de préparer le lecteur au contenu des graphiques qui sont présentés dans cette partie. Les séparateurs SVM sont affichés de la manière suivante :

- les points d'apprentissage utilisés pour la construction du séparateur sont affichés avec deux symboles différents selon leur classe :
  - un cercle (bleu) pour les points en défaillance ;
  - une croix (rouge) pour les points en sûreté ;
- la frontière entre les classes est en trait plein noir ;
- la marge entourant la frontière est comprise entre les pointillés plus clairs ;
- les lignes de niveaux sont affichées en niveaux de gris ;
- les points supports ou vecteurs supports sont les points d'accroche de la frontière et sont représentés par des points entourés, où l'épaisseur de ce contour correspond au poids du point en question (la valeur de  $\alpha$  issue de l'optimisation : cf. équation II.15).

### III.3 La méthode de J.E. Hurtado - SVM(H)

L'idée de l'algorithme provient du livre de J.E. Hurtado sur les méthodes d'apprentissage statistique au service des études de fiabilité des structures [Hur04b]. Cet ouvrage a été l'élément initiateur de la thèse, soulevant un intérêt particulier pour l'utilisation des SVM dans le calcul de fiabilité. Le principe est simple : il s'agit de considérer les simulations de Monte-Carlo comme un exercice de classification. L'objectif est alors de construire une formulation analytique de la frontière de l'état-limite pour séparer les données de la simulation. Dans un ensemble de réalisations des variables de base, l'objectif est de comptabiliser le nombre de réalisations dans le domaine de défaillance.

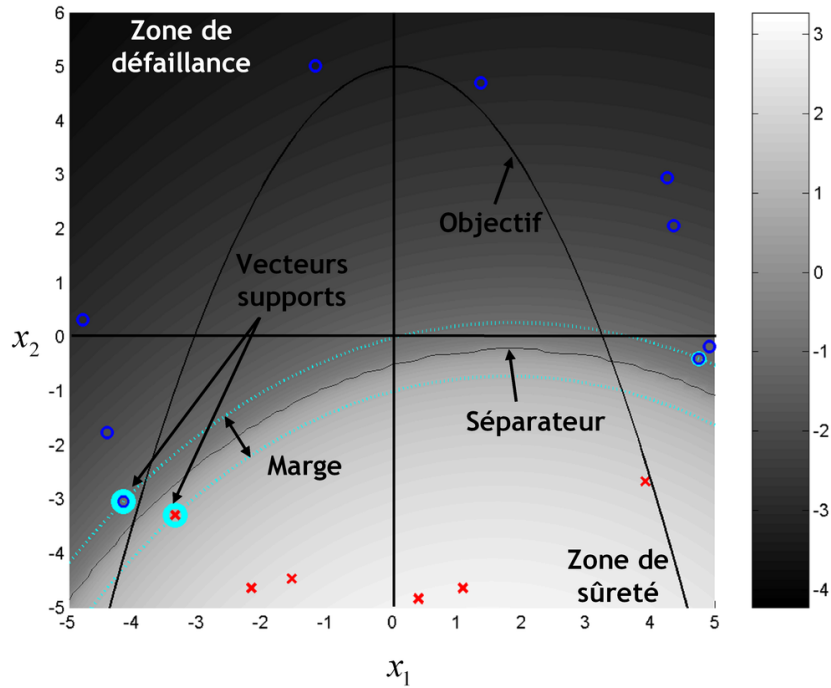
Le principe de l'algorithme de J.E. Hurtado, référencé par SVM(H), est présenté dans cette section. Cet algorithme a permis d'une part de prendre conscience de l'apport des SVM dans le calcul de fiabilité et de se familiariser avec les techniques et, d'autre part, de dégager des voies d'amélioration pour le calcul de fiabilité par la classification. Ces améliorations ont fait l'objet d'une communication [DL06].

#### III.3.1 Le principe de l'algorithme

Le principe de l'algorithme a tout d'abord été présenté dans un article de J.E. Hurtado [HA03]. Il est basé sur une propriété intéressante des SVM : pour améliorer le séparateur défini, il faut ajouter à la base d'apprentissage des points situés dans la marge. Ainsi à chaque itération, soit à chaque amélioration du séparateur, "quelques" points de la marge sont ajoutés à la base de données des points d'apprentissage. Cette propriété permet d'affiner le séparateur. J.E. Hurtado procède ainsi à l'amélioration du séparateur jusqu'à ce que celui-ci sépare parfaitement l'ensemble des réalisations de la simulation de Monte-Carlo. Le gain en temps de calcul est obtenu par le fait qu'un nombre restreint de réalisations permet de définir le séparateur SVM, qui permet quant à lui de séparer l'ensemble des points de la simulation et d'aboutir ainsi à la probabilité de défaillance.

#### III.3.2 Le détail pas à pas [Hur04a]

1. Transformation des variables de base  $X_i$  dans l'espace standard  $U_i$  (normé, centré et décorrélé). Cette étape est commune à toutes les méthodes de résolution du problème de la fiabilité. Elle permet de se placer dans un cas de symétrie



**Fig. III.3** Un premier séparateur est calculé sur la base d'apprentissage initiale constituée des points les plus éloignés de l'origine des simulations. Cette figure présente ce premier séparateur, ses caractéristiques - points supports, marge - et sa position par rapport à l'état-limite objectif.

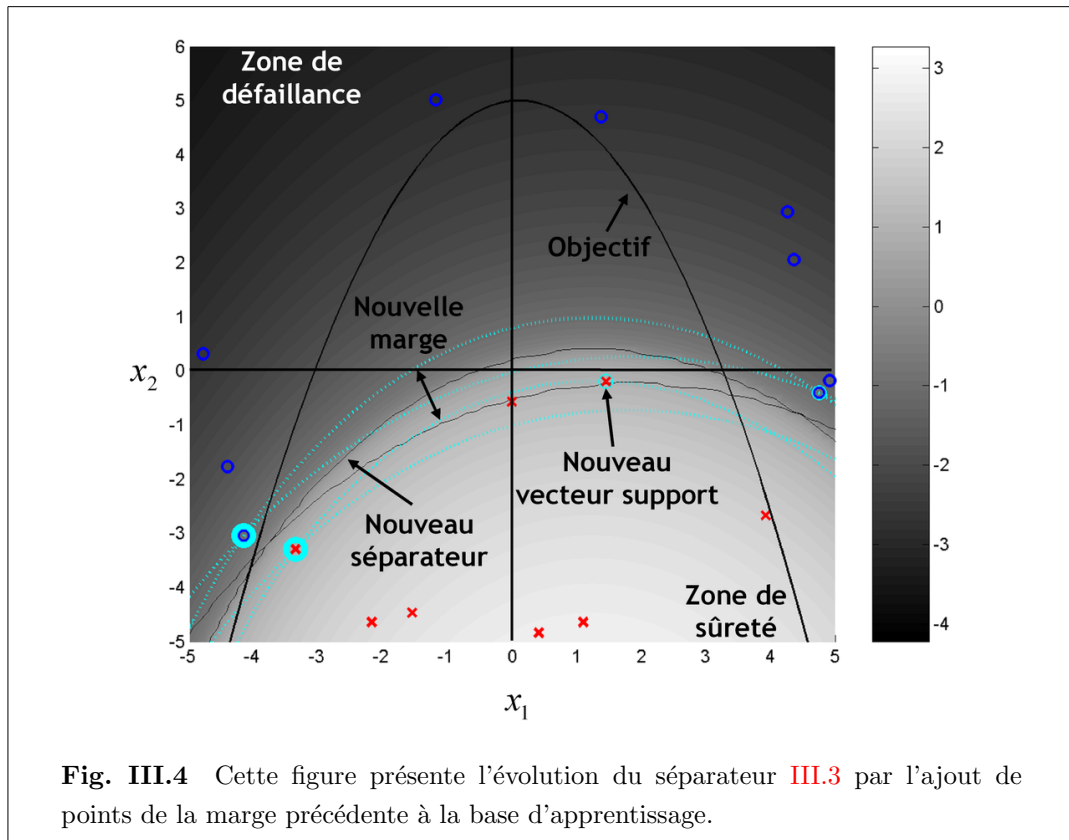
de révolution de l'espace de travail. Toutes les variables de base sont transformées en variables normales normées, centrées et également décorréliées. C'est une étape de préparation commune à la majeure partie des méthodes fiabilistes. (Cas des transformations de Nataf et de Rosenblatt - cf. I.4.)

2. Génération de  $N$  réalisations  $U_i$ ,  $i = 1, \dots, N$  selon la fonction de densité conjointe des variables (cette étape est simplifiée par l'étape précédente qui a permis de décorréler les variables aléatoires de base et de les transformer en variables normales indépendantes, centrées et normées). Ces  $N$  réalisations constituent l'ensemble des points à classer pour obtenir une approximation satisfaisante de la probabilité de défaillance. (Rappelons qu'une précision satisfaisante d'une probabilité de l'ordre de  $10^{-n}$ , nécessite  $10^{n+2}$  à  $10^{n+3}$  réalisations.)
3. Sélection des  $k$  réalisations les plus éloignées du centre des simulations. Le but est ici d'obtenir, parmi les  $k$  premiers points d'apprentissage, au moins une réa-

lisation de chaque classe. Ceci est indispensable au démarrage de l'algorithme. La connaissance experte du comportement du système peut aider à une optimisation de cette première étape. J.E. Hurtado considère qu'en prenant les points les plus éloignés, la classe de défaillance est atteinte.

4. Évaluation de la fonction de performance pour les  $k$  simulations sélectionnées. Ce sont les premiers appels à la fonction de performance  $G$ . La réponse du système va permettre d'associer aux points d'apprentissage la classe "+1 " sûreté ou "-1 " défaillance.
5. Calcul du séparateur sur les  $k$  points d'apprentissage. La résolution du problème d'optimisation II.15 permet d'obtenir la formulation analytique d'un premier séparateur  $\hat{G}(X)$ , et donc la définition de la marge initiale. La figure III.3 propose une représentation graphique du premier séparateur.
6. Recherche, parmi les  $N$  réalisations, de celles appartenant à la marge, et sélection du point le plus proche du séparateur. La distance de chacun des  $N$  points simulés au séparateur est obtenue par évaluation de  $\hat{G}$ . Le point le plus proche de la frontière est sélectionné et ajouté à la liste des points d'apprentissage.
7. Évaluation de la fonction de performance pour le point supplémentaire sélectionné et recherche du nouveau séparateur (et de sa marge) pour la liste de points d'apprentissage augmentée d'une unité. Cette étape permet l'obtention d'une nouvelle frontière et d'une nouvelle marge. La figure III.4 propose l'évolution du séparateur par l'ajout du point d'apprentissage issu de la marge précédente.
8. Retour aux étapes 6 et 7 tant qu'il y a des points dans la marge. L'absence de points dans la marge met fin à l'algorithme. Le séparateur obtenu est donc une formulation explicite de l'état-limite. La classification de l'ensemble des  $N$  réalisations à l'aide de ce dernier permet une évaluation de la probabilité de défaillance. Le résultat est plus précis que celui issu d'un calcul FORM ou SORM, puisque l'état-limite est approximé globalement et non pas localement autour du point de défaillance le plus probable.

Il faut noter que les marges successives ne sont pas (hélas) incluses les unes dans les autres.



### III.4 Discussion et propositions d'améliorations de l'algorithme

Le but de la démarche proposée est une sélection de points d'apprentissage garantissant une convergence vers le séparateur de référence (séparateur objectif) plus rapide que celle utilisée par J.E. Hurtado afin de limiter le nombre d'évaluations de la fonction d'état-limite. La démarche s'appuie sur les deux phases suivantes :

1. obtention d'une approximation explicite de l'état-limite par apprentissage. Les SVM aboutissent à une formulation explicite de la frontière entre le domaine de défaillance et le domaine de sûreté ;
2. réalisation de simulations de Monte-Carlo sur cette frontière explicite, afin d'obtenir une évaluation de la probabilité de défaillance. Cette seconde étape est "très rapide" puisque la frontière est obtenue analytiquement. La probabilité de défaillance du système est estimée comme le rapport entre le nombre de réalisations dans le domaine de défaillance et le nombre total de réalisations classées.



L'originalité de l'algorithme proposé consiste à dissocier le travail de construction du séparateur et le calcul de la probabilité de défaillance par Monte-Carlo.

Ces améliorations optimisent l'apprentissage et donc la sélection des points au démarrage et lors des itérations successives. Ces "points d'apprentissage" correspondent aux points où la fonction d'état-limite est évaluée. C'est en améliorant ce processus de sélection de points que l'algorithme montre un réel intérêt : nos efforts se concentrent donc sur la création d'une base de données optimale.

Autrement dit, les points clés sont :

- le premier plan d'expériences, c'est-à-dire la phase d'initiation de l'algorithme ;
- la sélection des points d'apprentissage dans les étapes d'amélioration de la classification.

L'objectif, en terme de résultat, est d'obtenir la probabilité de défaillance exacte, à comparer avec le résultat de Monte-Carlo considéré comme référence.

Lors des tests, l'algorithme de fiabilité FORM est réalisé sur les différents séparateurs obtenus. Cette analyse permet de comparer les points de conception obtenus sur le séparateur par rapport à la référence, et donc la précision obtenue dans les zones à fortes densités de probabilité.

#### III.4.1 La fonction noyau

Avant de détailler les points améliorés, donnons quelques informations concernant le type de la fonction noyau utilisée. Il est d'abord à noter que pour un exercice de classification, le choix de la fonction noyau n'influence que peu le résultat. La souplesse des différentes fonctions conduit à l'obtention de résultats similaires quel que soit le choix de l'utilisateur. Cependant, ceci est valable dans le cas où les données sont nombreuses. Dans notre cas, la règle est donc de fixer une fonction qui soit assez souple au regard du type de comportement à modéliser, sans oublier que le critère primordial reste la capacité de généralisation. Nous optons donc, dans le cas des SVM, pour les fonctions *Radial Basis Function (RBF)*, qui sont les fonctions les plus utilisées dans le contexte des méthodes de l'apprentissage statistique au service de la mécanique.

Ce choix se justifie par plusieurs raisons :

- d'une part, ce type de fonctions permet de modéliser des comportements fortement non linéaires. Rappelons que l'intérêt d'une méthode basée sur les SVM est justement d'apporter une réponse quant aux difficultés rencontrées par les surfaces de réponses quadratiques face à ces comportements non linéaires ;

- d'autre part, ces fonctions sont également majoritairement sélectionnées par les utilisateurs de réseaux de neurones en tant que fonction d'activation ;
- enfin, elles font également partie des fonctions utilisées lors de régression par la méthode du krigeage, qui est également une réponse à l'approximation de comportements mécaniques complexes [CGS03].

La sélection des fonctions noyaux n'est pas primordiale pour le problème de classification. Elle peut par contre le devenir lors d'un exercice de régression.

### III.4.2 Le plan d'expériences initial - carré latin

#### La problématique

Les variables aléatoires sont transformées dans l'espace standard. Intuitivement, le plan initial, *i.e.* les premiers points dont la classe est évaluée, va conditionner la suite de l'algorithme. Toutefois, la marge du séparateur évolue et les marges successives ne sont pas contenues dans la marge initiale. Par conséquent, un mauvais placement initial du séparateur ne conduit pas nécessairement à l'échec de l'algorithme mais ralentit la vitesse de convergence. Deux objectifs sont associés au plan initial :

- le premier est d'obtenir des réalisations dans les deux domaines de l'espace : sûreté et défaillance. La probabilité de défaillance calculée étant faible, il est *peu probable*, par un tirage aléatoire, d'obtenir des points dont l'évaluation de l'état-limite est négative ;
- le second objectif est d'avoir une connaissance assez globale de l'espace pour optimiser la vitesse de convergence de l'algorithme.

Pour garantir ces objectifs plusieurs techniques sont proposées.

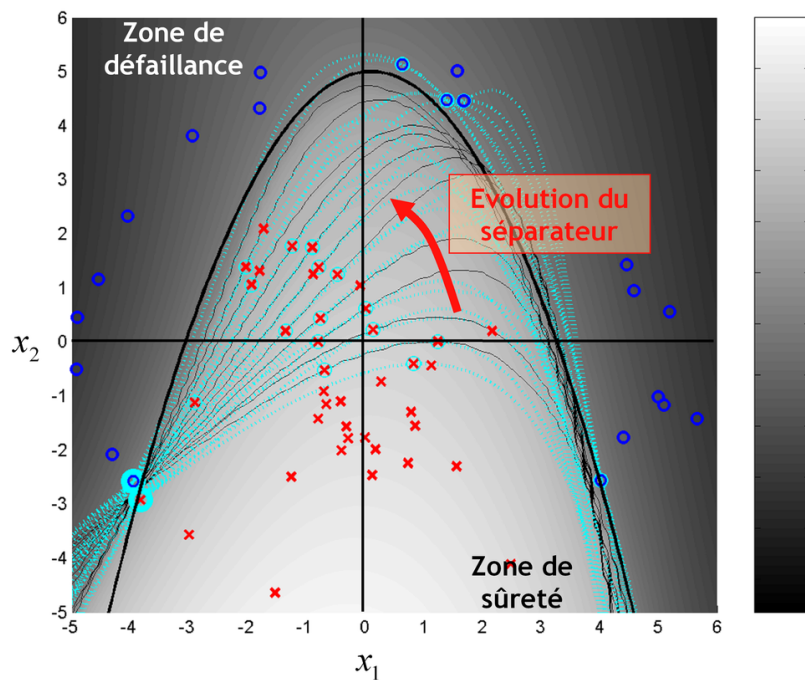
#### Le choix de J.E. Hurtado

J.E. Hurtado propose, pour démarrer l'algorithme, de sélectionner les points les plus éloignés d'une simulation afin d'en obtenir au moins un en zone de défaillance. Ceci est valable lorsque l'on part du principe qu'il faut 2 points pour démarrer l'algorithme : un dans chaque catégorie. Malheureusement cela nous conduit au calcul de nombreux points inintéressants et inutiles.

Sur la figure III.5, la construction du séparateur a été initialisée avec les points les plus éloignés du centre des simulations (c'est-à-dire de l'origine du repère dans l'espace standard). Étant donné le manque d'information autour de l'origine, l'algorithme a nécessité plusieurs itérations pour s'approcher de la "forme" recherchée. Les points

calculés dans les marges des premiers séparateurs obtenus sont devenus inutiles par la suite, car trop éloignés des zones d'intérêt. La lenteur de l'algorithme est évidente.

Si le positionnement du premier séparateur est loin de la position qu'il doit atteindre, un grand nombre d'itérations est nécessaire et de nombreuses évaluations de l'état-limite sont inutiles. Ce point met en exergue la nécessité d'optimiser le plan servant au démarrage de l'algorithme.



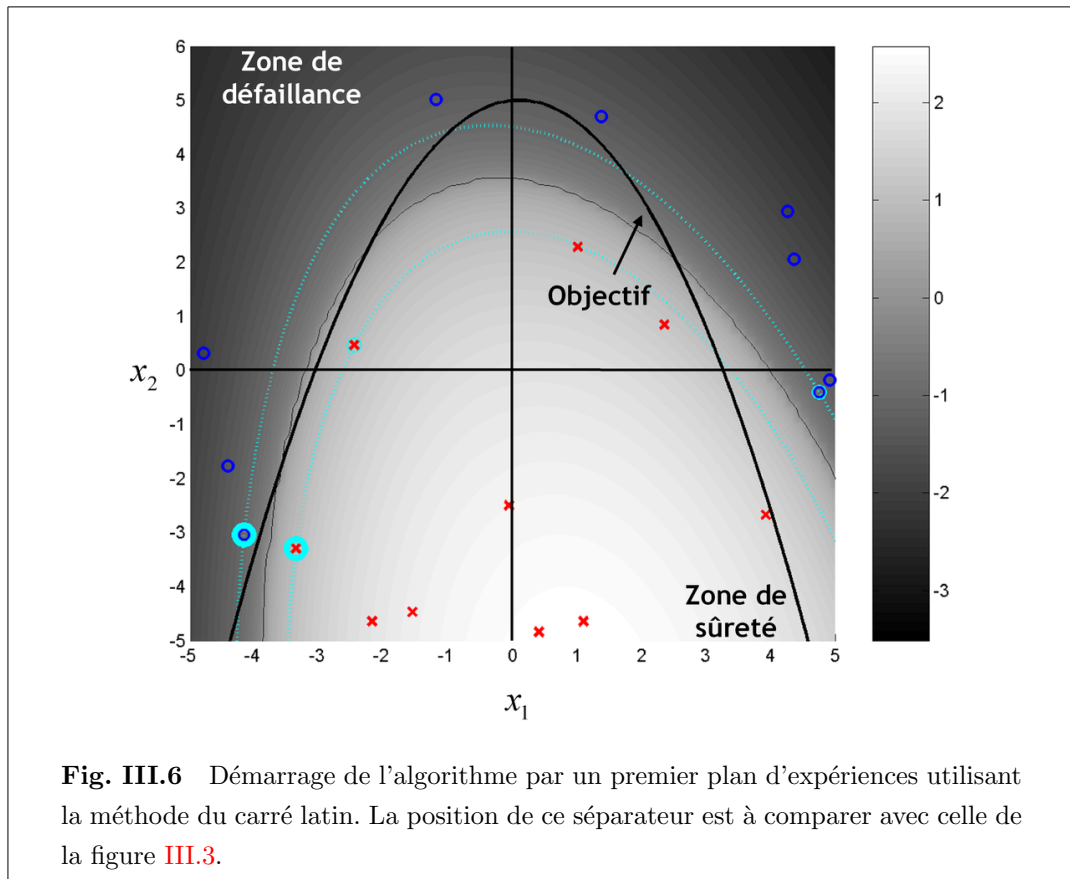
**Fig. III.5** Les itérations du processus de construction du séparateur par la méthode de J. Hurtado. Les premiers points d'apprentissage sont les points de la simulation les plus éloignés de l'origine. De nombreuses itérations du processus sont nécessaires pour un positionnement satisfaisant du séparateur.

### Le carré latin

La méthode proposée consiste à utiliser les plans d'expériences pour l'exploration de l'espace et pour l'obtention d'une approximation de "forme" plus rapidement proche de celle qu'on souhaite atteindre. Le plan d'expériences, parce qu'il est issu des théories statistiques, est probablement la meilleure solution pour démarrer l'algorithme avant d'utiliser les propriétés des SVM et de la marge du séparateur pour

affiner l'approximation. Pour conserver une faible dépendance aux nombres de variables aléatoires, les plans d'expériences complets ( $2^n$ ,  $3^n$  ...) sont abandonnés et pour conserver une faible corrélation entre plan d'expériences et résultat, nous optons pour les plans aléatoires.

Le démarrage de l'algorithme utilise un nombre donné de points simulés selon la méthode du carré latin I.7.4 dans un hypercube de côté  $\beta_{\max}$  ou dans une hypersphère de rayon équivalent, distance fixée en fonction du niveau de fiabilité attendu et de la dimension de l'espace. Cette méthode est recommandée pour la régression par la méthode du krigage, plus proche de la régression SVM que la méthode des moindres carrés avec surface quadratique, pour laquelle les plans réguliers sont utilisés et recommandés [NIS06].



La figure III.6 montre l'initialisation de l'algorithme. Le plan d'expériences balaie l'espace de départ de manière aléatoire et améliore le premier positionnement du séparateur. Le carré latin garantit une couverture uniforme de tout l'espace avec un

minimum de points.

### Plan d'expériences expert

Certains problèmes possèdent des caractéristiques connues par le concepteur. La connaissance experte peut apporter une précision sur la répartition de la défaillance dans l'espace ou les zones d'importance. Il est dans ce cas important de pouvoir prendre en compte ces connaissances.

L'approche consiste alors à mixer un plan étoilé expert (dimension  $2N + 1$ ) et quelques points d'un plan de type carré latin. Le premier va permettre, en se plaçant dans les queues de distribution des lois des variables du problème, d'obtenir des réalisations dans la zone de défaillance. Le second permet d'ajouter de la connaissance sur des zones choisies aléatoirement dans l'espace de travail.

Pour des probabilités de défaillance très faibles, l'obtention de points dans le domaine de défaillance est très difficile. Pour contrer ce problème, en évitant soit d'augmenter le nombre de points dans le plan initial (lourd en terme de calcul), soit d'élargir l'hypercube dans lequel est tiré le plan initial (problème de validité du modèle mécanique), une autre méthode est proposée. Il s'agit d'une utilisation ponctuelle de la régression. Dans le cas où les points du plan initial ne permettent pas de commencer la classification, une régression est effectuée sur ces premiers points. La surface obtenue permet de proposer une zone de l'espace dans laquelle il est très probable d'être proche de la zone de défaillance. Cette méthode permet ainsi de délimiter la zone d'intérêt qui est la zone proche de l'état-limite. La régression permet de prendre en compte une information à disposition qui était jusque là non utilisée. En effet, la classification utilise uniquement l'information binaire  $+1$  et  $-1$  de l'évaluation de l'état-limite.

### Bilan

Le carré latin est retenu pour la première étape de l'algorithme. Il reste cependant quelques paramètres à ajuster :

- le nombre de points simulés ;
- la taille du rayon de l'hypercube.

Ces paramètres sont liés à la probabilité de défaillance cible et à la dimension du problème. Il convient d'ajuster ces valeurs. En effet, il faut simuler assez de points pour ne pas avoir de trop nombreuses itérations du processus d'amélioration, mais

bien entendu pas trop de points, car les appels à l'état-limite sont comptés. Le nombre de simulations est en relation avec la méthode d'amélioration du séparateur, présenté dans la section suivante. En effet, il est possible d'imaginer différentes stratégies d'apprentissage en fonction de la précision escomptée sur le résultat et du nombre de points que l'on s'autorise à calculer pour la base d'apprentissage.

### III.4.3 Choix des points d'apprentissage dans la marge

#### La problématique

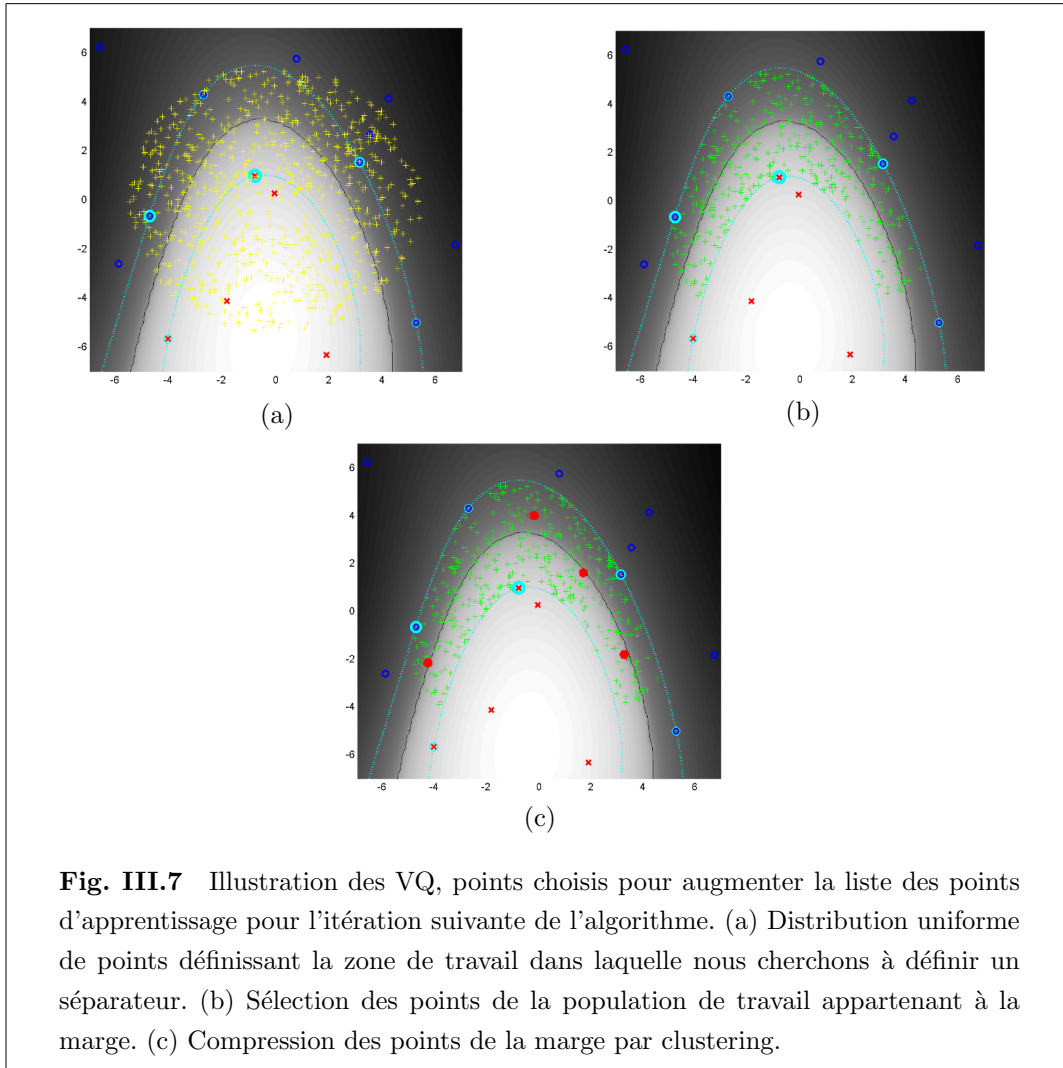
Le choix des points d'apprentissage est crucial, puisque la frontière en dépend. La méthode qui va permettre de sélectionner les points de l'apprentissage est donc très importante. Les SVM nous donnent une information primordiale : pour affiner le séparateur, il faut choisir des points dans la marge. L'objectif est d'évaluer un minimum de points en affinant au mieux la marge dans la zone de l'espace qui correspond à une zone de travail. Cet espace, pour J.E. Hurtado, correspond à l'ensemble des données à classer. En dissociant le problème de l'apprentissage et du calcul de la probabilité, nous avons perdu cette notion primordiale. Pour appliquer ce principe, les points d'apprentissage sont choisis dans une liste de simulations dans une hypersphère de rayon  $\beta_{\max}$ , simulation construite arbitrairement pour constituer une zone d'apprentissage. Seuls les points de cette *population de travail* appartenant à la marge du séparateur sont retenus pour être des candidats à la liste des points d'apprentissage.

#### Le choix de J.E. Hurtado

J.E. Hurtado sélectionne "LE" point le plus proche du séparateur parmi les points de la simulation à classer. En théorie, cette technique permet, au bout d'un certain nombre d'itérations, d'obtenir un séparateur correct. Mais, dans les applications initiales effectuées, cette technique montre une première limite : de nombreux points calculés ne sont pas utiles et l'algorithme avance petit à petit, comme le montre la figure III.5 sur l'évolution du séparateur. Ceci est vrai surtout si le nombre de simulations de la liste est grand. Une amélioration plus globale et plus rapide du séparateur est aussi souhaitable.

#### Compression de distribution

Nous choisissons de sélectionner, parmi les points de la distribution de travail, ceux appartenant à la marge. Après le calcul d'un séparateur, la marge est défi-



**Fig. III.7** Illustration des VQ, points choisis pour augmenter la liste des points d'apprentissage pour l'itération suivante de l'algorithme. (a) Distribution uniforme de points définissant la zone de travail dans laquelle nous cherchons à définir un séparateur. (b) Sélection des points de la population de travail appartenant à la marge. (c) Compression des points de la marge par clustering.

nie. En sélectionnant l'intersection de la marge et des points de la simulation, nous obtenons un ensemble de points dans lequel nous allons choisir nos points d'apprentissage. L'idée est de sélectionner "quelques" points répartis uniformément dans la marge. L'utilisation de la technique de *clustering* II.6.3 est apparue comme une solution satisfaisante. Cette technique permet de retenir quelques points de la liste qui représentent la simulation : ils en gardent les caractéristiques principales. Ces outils répondent parfaitement à nos besoins, puisque nous souhaitons avoir une information sur le comportement du séparateur dans la marge, et ceci de manière globale : sur toute la "longueur" de la marge.

Un aperçu de la compression par les VQ et de la sélection des points de la marge

est présenté dans la figure III.7. La figure (a) présente la population de travail, zone dans laquelle le séparateur est recherché. Cette zone peut correspondre au domaine de validité de la fonction de performance. La figure (b) présente la sélection des points de la population de travail qui appartiennent à la marge du séparateur. C'est uniquement dans cette zone que des points d'apprentissage sont utiles à l'amélioration du séparateur. La figure (c) présente la compression des points de la marge par clustering. La répartition des points le long de la marge est assez homogène. Cette illustration montre la capacité des VQ à représenter une distribution non régulière : la distribution des points de la marge. Nous retenons cette technique pour le choix des points d'apprentissage dans la marge.

#### III.4.4 Approche multi-échelle

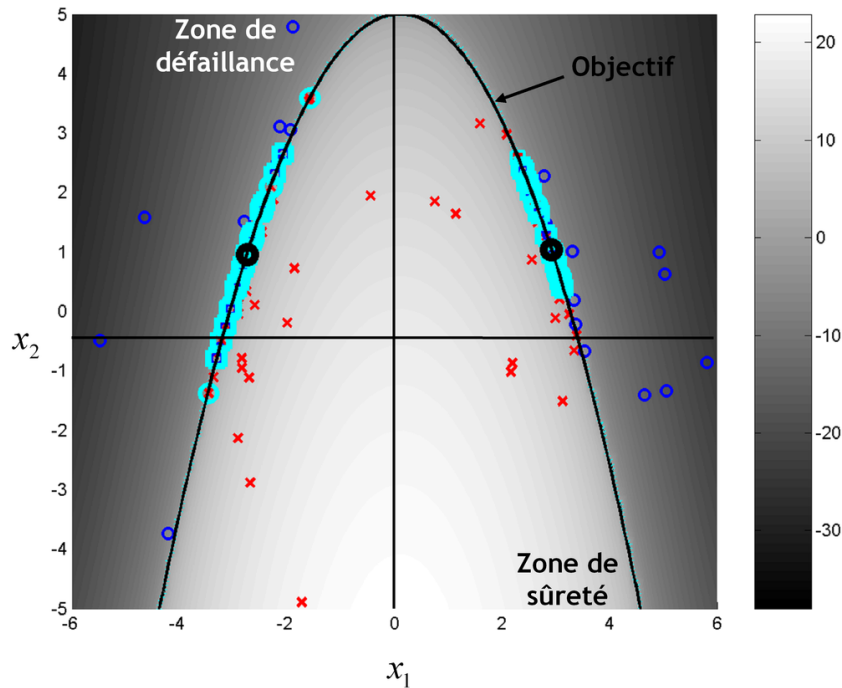
##### La problématique

Dans le paragraphe précédent a été introduite *la notion d'espace à classer*, c'est-à-dire une zone de l'espace dans laquelle nous cherchons des informations sur notre système. Cet espace est délimité par la simulation dans laquelle nous allons sélectionner des points d'apprentissage. L'amélioration du séparateur se fait jusqu'à obtention d'une marge vide. À partir de ce moment, les points de la simulation sont classés par notre frontière analytique.

En partant de ce principe, nous décidons de procéder par étapes successives pour accélérer la classification. Chaque étape correspond à la classification d'un ensemble de plus en plus dense de points : nous parlons d'approche multi-échelle.

Le but de l'algorithme est de parvenir à classer un ensemble de  $10^{n+2}$  à  $10^{n+3}$  points (pour une probabilité de défaillance de l'ordre de  $10^{-n}$ ), points correspondant à une simulation de Monte-Carlo des variables de base dans l'espace standard. Mais si l'on travaille directement avec cette simulation, la procédure d'apprentissage et de classification est lente. En effet, l'algorithme est plus rapide avec une population moins importante, même si cela conduit à une précision moindre. De tels résultats incitent à séparer l'algorithme en plusieurs étapes, afin d'obtenir plus rapidement un séparateur proche de la forme finale, puis d'affiner autour des zones d'intérêt, *i.e.* les zones à plus forte densité de probabilité.





**Fig. III.8** Illustration de l'utilité de sélectionner les points d'apprentissage dans une liste de réalisations simulées selon une gaussienne autour de l'origine de l'espace standard.

### Des étapes d'apprentissage successives

Une première étape du travail d'amélioration est réalisée sur une simulation uniforme de  $N_1$  points (de l'ordre de  $10^3$  à  $10^4$ ) autour de l'origine. L'amplitude de cette distribution dépend de la taille de l'espace et également de la probabilité de défaillance cible. En classant ces premiers points est obtenue une première version du séparateur : la forme est peu précise mais sa position dans l'espace est proche de celle de l'état-limite recherché. Cette première étape se fait en une dizaine d'itérations du procédé.

Dans une seconde étape, nous cherchons à améliorer la classification dans les zones à fortes densité de probabilité. Pour cela, nous allons utiliser une simulation gaussienne de  $10^{n+3}$  points. Les raisons de cet emploi sont les suivantes :

- augmentation du nombre de points afin d'augmenter la précision du séparateur ;
- choix d'une gaussienne afin qu'il y ait plus de points dans les zones à forte densité de probabilité : les VQ étant pondérés par la densité, le processus est

amélioré naturellement autour des points de conception qui concentrent la plus importante partie de la probabilité de défaillance.

La figure III.8 illustre le fait que les points sélectionnés pour augmenter l'apprentissage sont naturellement concentrés autour des zones à forte densité de probabilité : c'est-à-dire les zones de défaillance les plus proches de l'origine.

À chaque itération, la simulation est générée à nouveau afin de travailler sur une population nouvelle à chaque fois. Cela permet de perturber l'algorithme et d'éviter un biais éventuel. Cette approche multi-échelle peut être affinée au besoin en insérant des étapes supplémentaires. Cette technique cherche à éviter le calcul de points qui seraient inutiles par la suite.

#### III.4.5 Bilan : trois phases d'apprentissage

Après différents tests, trois phases d'apprentissage sont dissociées. À chacune de ces phases correspond une population de travail, dans laquelle des points sont sélectionnés pour enrichir la base d'apprentissage.

1. *Phase de positionnement* : la première étape de l'apprentissage se fait sur une population répartie uniformément dans la zone de définition de la fonction d'état-limite. Cette population est peu dense (estimation  $10^3$  à  $10^4$  individus en fonction de la taille du problème). La sélection des points de calculs mécaniques se fait par *clustering* des points de la marge.
2. *Phase de stabilisation* : la seconde étape de l'apprentissage se fait également sur une population répartie uniformément, mais cette dernière est plus dense : cette étape permet une stabilisation de la position du séparateur et également une amélioration de la précision globale par rapport à l'état-limite objectif (estimation  $10^4$  à  $10^6$  individus en fonction de la taille du problème). La sélection des nouveaux points se fait par *clustering* des points de la marge, ainsi que des points que l'on appelle instables, ie. qui changent de côté entre deux itérations.
3. *Phase d'amélioration de la précision* : cette dernière étape se fait sur une population gaussienne de la taille de la simulation de Monte-Carlo prévue. Cela permet d'une part d'affiner le séparateur uniquement dans la zone nécessaire à l'évaluation de la probabilité de défaillance. La population gaussienne permet également de favoriser la sélection de nouveaux points dans les zones de forte densité de probabilité de défaillance. Les nouveaux points sélectionnés sont les points les plus proches de l'état-limite.

### III.4.6 Critères de validation

La partie qui demande le plus d'attention lors de la construction d'une méthode numérique, est celle consacrée à la convergence de l'algorithme et également à la mesure de l'erreur. Dans le meilleur des cas, il est possible concevoir précisément l'erreur obtenue. Si une telle donnée est impossible à mesurer, l'obtention d'un intervalle de confiance ou encore de bornes sur le résultat est très utile voire indispensable.

Dans la méthode proposée, la convergence de l'algorithme est basée sur une stabilisation du résultat fiable, calculé par simulations de Monte-Carlo sur la surface explicite obtenue. Cela ne garantit pas forcément l'obtention du résultat optimal. En effet, cette mesure dépend de la qualité de la surface obtenue par les SVM et de sa stabilité vers la surface objectif. Si on ne peut garantir que la frontière obtenue est proche de l'objectif, certains indices permettent de crédibiliser le résultat. La marge permet notamment de définir différents critères de validation : la taille de la marge, la valeur de l'état-limite obtenue sur les points d'apprentissage successivement ajoutés, la valeur de l'état-limite des vecteurs supports...

D'autres pistes sont proposées, pistes permettant de mesurer la stabilisation, au sens de la position dans l'espace, de la surface obtenue. Les mettre en place nécessite de conserver les surfaces obtenues à chaque itération du processus, pour permettre de mesurer l'évolution de différentes quantités.

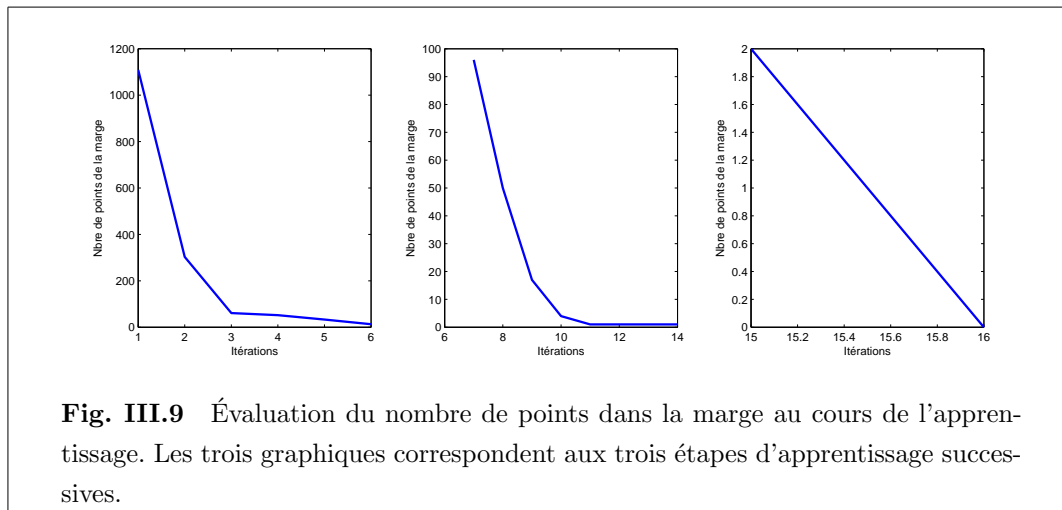
Ces différents critères sont illustrés sur le cas test, fil conducteur du chapitre.

#### Critères sur la marge

Différents critères sont proposés à partir de l'évolution de la marge au cours des itérations du processus de construction du séparateur. Le concept de marge permet non seulement de générer les futurs points d'apprentissage, mais il peut également permettre de vérifier la convergence de la méthode. La population de travail étant renouvelée à chaque étape pour optimiser l'apprentissage, une population de référence est utilisée pour évaluer les critères définis ci-après.

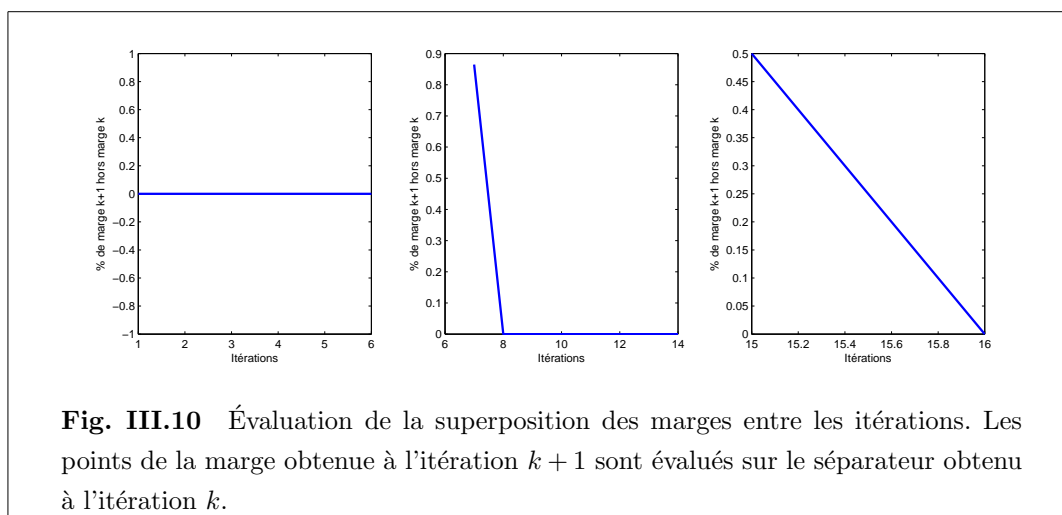
1. *Le nombre de points dans la marge.* A chaque itération de l'algorithme, une sélection de points de la marge est faite. Pour cela, une simulation de points est évaluée sur la fonction SVM. Cette évaluation permet de définir un ensemble de points appartenant à la marge, et permet donc le comptage des points situés dans cette marge. Si ce nombre de points tend vers 0, alors l'ensemble des points de la simulation est classé par le séparateur. Une diminution de cette valeur

montre que la marge s'affine.



**Fig. III.9** Évaluation du nombre de points dans la marge au cours de l'apprentissage. Les trois graphiques correspondent aux trois étapes d'apprentissage successives.

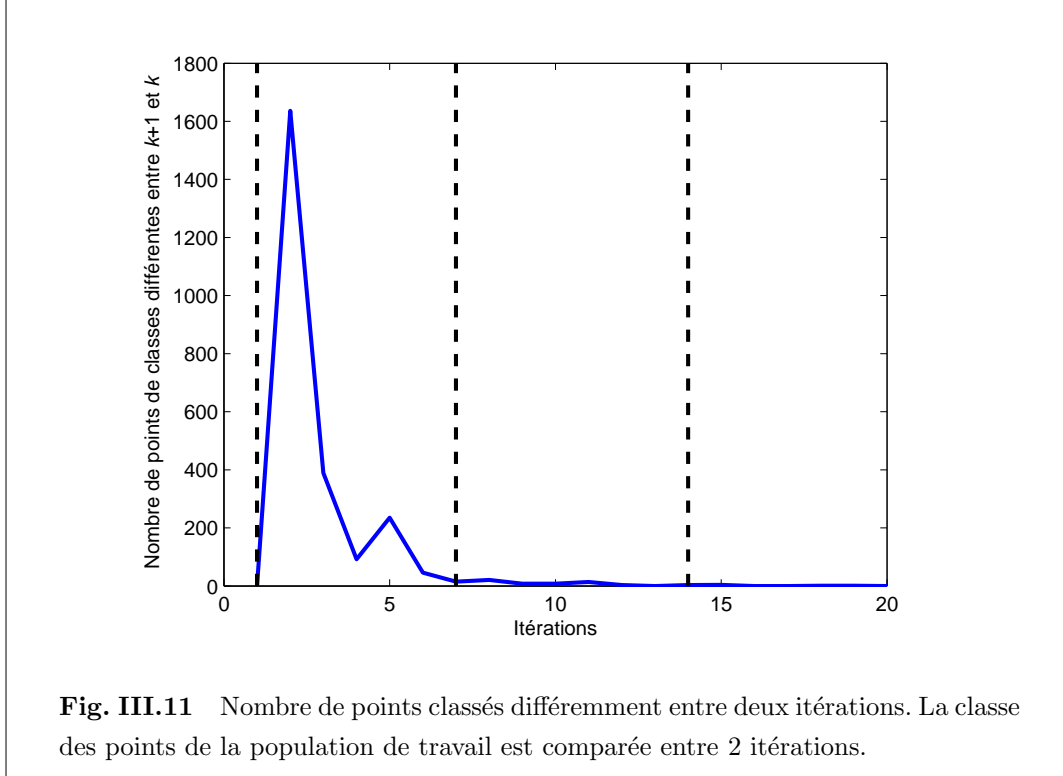
Sur la figure III.9, 3 courbes sont affichées. Elles correspondent aux différentes simulations effectuées pour la recherche de nouveaux points d'apprentissage, étapes détaillées dans la section consacrée à l'approche dite multi-échelle III.4.4. Pour chaque population de travail, le nombre de points situés dans la marge diminue au cours des itérations d'apprentissage.



**Fig. III.10** Évaluation de la superposition des marges entre les itérations. Les points de la marge obtenue à l'itération  $k+1$  sont évalués sur le séparateur obtenu à l'itération  $k$ .

2. Le pourcentage de points de la marge  $k+1$  en dehors de la marge  $k$ . Ce pourcentage est une certaine mesure de la stabilité des séparateurs. A chaque itération, la superposition des marges est vérifiée : la marge obtenue est comparée à la

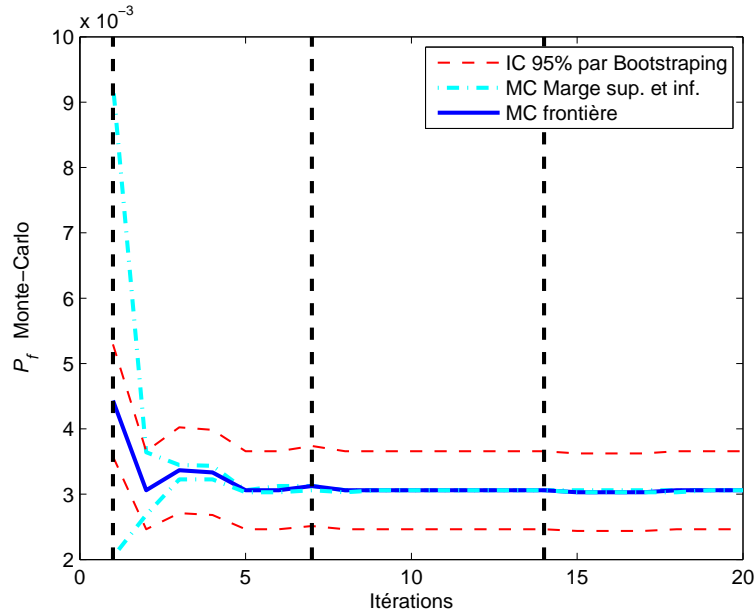
marge précédente. Un pourcentage très petit montre que la marge  $k+1$  est comprise dans la marge  $k$ . La figure III.10 présente le type de graphique obtenu.



3. *Nombre de points classés différemment.* Les deux premiers critères permettent de vérifier la stabilité de la marge dans l'espace de définition. Mais un troisième critère est plus intéressant, il compare les classes obtenues par les points de la simulation de Monte-Carlo. Entre l'itération  $k+1$  et  $k$ , le nombre de points de classes différentes est recensé. Ce critère permet de mesurer la précision du résultat en probabilité de défaillance. Le graphique obtenu pour le cas test est présenté figure III.11.

### Validation de la valeur de $P_f$

Plusieurs critères de validation sont élaborés par rapport à la valeur de la probabilité de défaillance.



**Fig. III.12** Évolution de la probabilité de défaillance par simulation de Monte-Carlo sur le séparateur au cours de l'apprentissage.

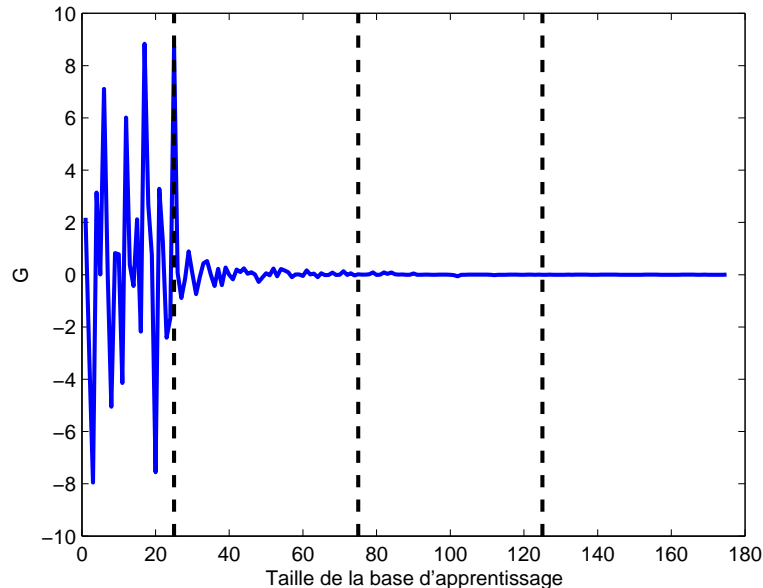
**(a) Intervalle de confiance.** Pour vérifier la stabilité des résultats par rapport à la sélection des points d'apprentissage, une technique de *bootstrapping* est utilisée. Une telle approche a été développée précédemment au laboratoire [GBL03]. La technique du *Jackknife* a été testée dans un premier temps. Cette technique est appliquée à chaque itération de la phase d'apprentissage. Cela permet d'obtenir un intervalle de confiance sur le résultat en probabilité de défaillance, calculé sur les séparateurs à chaque itération. Pour chaque séparateur, 20 nouveaux échantillons sont créés à partir des points d'apprentissage, par la méthode du *Leave-One-Out* (*Jackknife*). Sur chacun des échantillons est calculé un séparateur, sur lequel la probabilité de défaillance est évaluée. Cette technique fournit plusieurs valeurs de la probabilité de défaillance et une mesure statistique peut être effectuée. La moyenne et l'écart type sont évalués, aboutissant ainsi à un intervalle de confiance sur la valeur de la probabilité de défaillance. Cette technique est intéressante surtout dans le cadre de la régression : en fait les points d'apprentissage qui ne sont pas des points supports n'ont aucun poids dans la formulation du séparateur pour le cas de la classification. L'intervalle de confiance obtenu par *Jackknife* montre un résultat très stable qui ne reflète peut-être pas la stabilité réelle.

La figure III.12 montre l'évolution de la probabilité de défaillance calculée sur les séparateurs obtenus à chaque itération.

**(b) Monte-Carlo sur la marge.** Une information supplémentaire est disponible sur la figure III.12 :  $Pf_{marg\ sup}$  et  $Pf_{marg\ inf}$ . Si l'on considère que la marge correspond à la zone d'incertitude de la classification, il est intéressant de calculer la probabilité de défaillance associée à un état-limite qui correspondrait à la frontière inférieure de la marge ( $Pf_{marg\ inf}$ ) ou à la frontière supérieure de la marge ( $Pf_{marg\ sup}$ ). L'intervalle défini par ces valeurs se resserre autour de la valeur moyenne. Cela montre que le séparateur obtenu est de plus en plus précis au cours des itérations, mais cela ne correspond pas forcément à des bornes supérieure et inférieure absolues de la probabilité de défaillance.

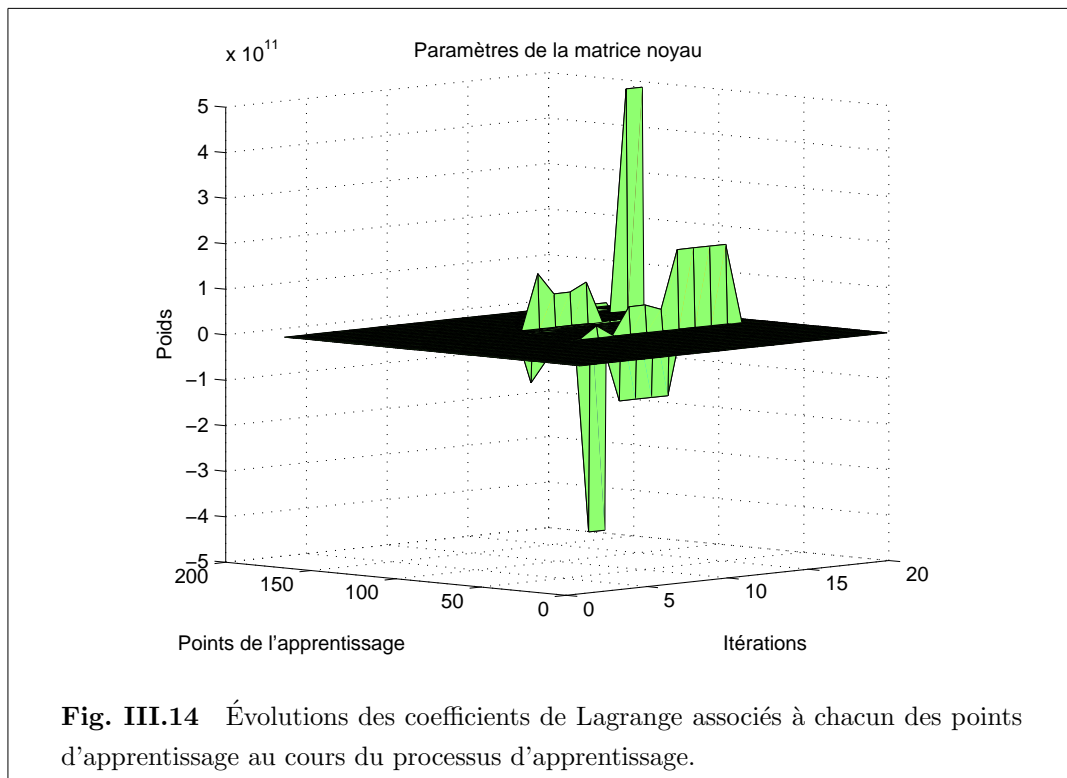
### Autres critères

D'autres mesures permettent de suivre l'évolution de l'algorithme.



**Fig. III.13** Évaluations de l'état-limite des points d'apprentissage. La convergence de la courbe vers 0 montre la propriété de l'apprentissage à s'affiner autour de l'état-limite objectif.

1. *Évaluations de l'état limite.* La figure III.13 montre que les points d'apprentissage, obtenus par la méthode de sélection des points de la marge, amènent le processus à s'affiner autour de l'état-limite, puisque les évaluations de celui-ci convergent vers 0.
2. *Évolutions des paramètres de la fonction SVM.* Il est simple de voir le poids affecté aux points de l'apprentissage. Les points ajoutés par la sélection dans la marge prennent un poids plus important que les points du plan d'expériences initial. Ils sont en fait plus proches de la marge, et sont tour à tour pris comme points supports (ou vecteurs supports) du séparateur. La stabilité du poids affecté à un point montre une certaine stabilité du séparateur, et l'importance du point dans la définition du séparateur, voir figure III.14.



## III.5 Bilan sur l'algorithme SMART

L'algorithme obtenu à partir des différentes améliorations de la méthode de J.E. Hurtado est nommé SMART pour *Support-vector Margin Algorithm for Reliability*



*esTimation*. Les différentes stratégies d'apprentissage retenues et les critères de validation permettant le suivi de la convergence en font un algorithme dédié à la recherche d'état-limite pour la fiabilité.

Cette section présente quelques remarques sur cet algorithme de construction du séparateur. Cela comprend des remarques par rapport aux résultats obtenus mais également des réflexions sur ce qu'il est utile d'apporter pour rendre la méthode plus souple et robuste.

	MC	FORM	SMART
$N$ appels	$10^6$	20	50
$P_f$ ( $10^{-3}$ )	3,0	1,8	3,0
cv	0,02	—	0,02

**Tab. III.2** Résultats du calcul de fiabilité sur le modèle du fil conducteur, en terme de probabilité de défaillance. La valeur de  $\gamma$  est fixée ici à 5. Il faut noter que dans cet exemple, FORM ne peut être satisfaisant.

	FORM	FORM sur SVM
$N$ appels	20	50
$\beta$	2,9	2,9
$u_1$	-2,741	-2,778
$u_2$	0,965	0,845

**Tab. III.3** Résultats du calcul de fiabilité sur le modèle du fil conducteur, en terme de point de défaillance le plus probable.

### III.5.1 Une vision globale

L'approximation globale obtenue est satisfaisante dans la zone de l'espace ayant un intérêt pour le calcul de la probabilité de défaillance : les résultats obtenus par simulation de Monte-Carlo sur la frontière explicite, sont proches des résultats de référence (cf. tableau III.2). Le séparateur obtenu donne une vision globale satisfaisante de la fonction à approximer.

L'approximation est également assez précise localement (au niveau des minima ou points de conception), au vu des résultats obtenus aux calculs FORM en terme d'indice de fiabilité (cf. tableau III.3), mais également en terme de point de défaillance

le plus probable, où la convergence est souvent plus difficile. Ceci est dû au fait que l'algorithme "pioche" des points dans la liste de réalisations simulées selon une gaussienne. La densité de points dans la marge du séparateur est plus importante autour du point de conception le plus probable, la simulation s'appuyant sur la distribution de la multi-normale.

### III.5.2 Une frontière explicite

La formulation d'une frontière explicite dont le comportement peut être fortement non linéaire et même discontinu (cas test de fiabilité système : union et intersection d'états-limites linéaires), constitue le point fort de la méthode. La souplesse face à la forme de l'état-limite et la non dépendance au plan d'expériences, en font une méthode très intéressante dans le contexte de l'analyse mécano-fiabiliste des systèmes complexes. Cette méthode permet également, à partir de résultats d'expérimentations numériques, de formuler explicitement un état-limite sur lequel nous pouvons travailler sans nouvel appel au modèle mécanique souvent lourd.

### III.5.3 Paramétrage de la méthode

Certains paramètres sont à régler en fonction du problème à traiter :

- le nombre de points pour initialiser l'algorithme ;
- le nombre de points d'apprentissage à ajouter à chaque itération ;
- la taille de la simulation de Monte-Carlo pour évaluer chaque séparateur.

Le nombre de points d'apprentissage (initial ou à chaque itération) dépend naturellement de la taille de l'espace. Ces paramètres s'ajustent facilement puisque la dimension de l'espace est une connaissance préalable. En revanche la taille de l'espace à balayer au démarrage est une donnée inconnue (sauf rares cas où la probabilité de défaillance visée est connue). Par conséquent, il est difficile de trouver la zone de défaillance. Une astuce est proposée : il s'agit de réaliser une régression sur les premiers points d'apprentissage, afin de déterminer les zones de probable défaillance. Mais si la probabilité de défaillance est extrêmement faible, le problème de l'évaluation de notre séparateur par Monte-Carlo se pose à nouveau.

### III.5.4 En conclusion

Les bases d'une méthode de calcul de fiabilité fondée sur une substitution de l'état-limite par un séparateur SVM ont été présentées. Cette méthode est inspirée

de celle de J.E. Hurtado mais présente de meilleures propriétés de convergence et de robustesse (qui sont étudiées en détail dans le chapitre [IV](#)). La clé étant de chercher à affiner le séparateur en le dissociant du travail d'évaluation de la fiabilité. Des indicateurs de convergence et de suivi de l'approximation ont été proposés pour observer l'évolution de l'algorithme.

L'évaluation d'une probabilité de défaillance faible reste cependant une limite. En effet, ce cas de figure influence le choix des paramètres à utiliser : la taille du plan initial et le nombre de simulations nécessaires à l'évaluation de la probabilité de défaillance. La section suivante montre comment les *subsets simulations* peuvent apporter leur contribution à l'évaluation de celle-ci.

## III.6 Subsets Simulations et Classification : méthode <sup>2</sup>SMART

A l'utilisation de la méthode SMART, les limites des simulations de Monte-Carlo réapparaissent : si la probabilité de défaillance est très faible, il faut un très grand nombre de simulations. De plus les paramètres de l'algorithme jouent un rôle trop important pour la robustesse de la méthode. Chercher de telles probabilités de défaillance conduit à chercher dans un espace, qui peut être de grande dimension, quelques points de défaillance loin de l'origine. Au et Beck [[AB01](#)] répondent à ce problème par l'utilisation de la méthode des *subsets simulations*.

Nous proposons ici de combiner *subsets simulations* et SMART en essayant de profiter des avantages des deux techniques.

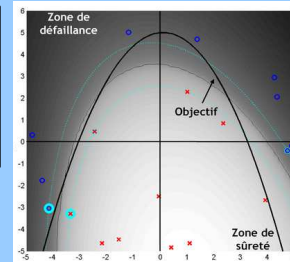
### III.6.1 Subsets simulations

Cette section présente la méthode des *subsets simulations*. L'objectif de la méthode est équivalent à celui de la simulation de Monte-Carlo classique, *i.e.* évaluer la probabilité de défaillance sans hypothèse sur la forme de l'état-limite. La méthode est particulièrement efficace lorsque la probabilité de défaillance est très faible, inférieure à  $10^{-3}$ - $10^{-4}$ , là où la méthode de Monte-Carlo devient trop coûteuse.

**Initialisation**

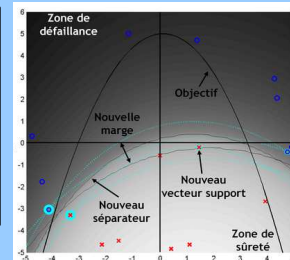
1. Plan d'expériences initial :
  - Carré Latin
  - Plan expert
2. Évaluations de l'état-limite.
3. Construction du premier séparateur.

**Objectif :** obtenir une répartition homogène des points dans l'espace, et au moins une réalisation dans les domaines de sûreté et de défaillance.



A chaque itération :

1. Recherche des points de la marge parmi la **population de travail**.
2. Sélection de points parmi les points de la marge.
3. Évaluations de l'état-limite.
4. Construction d'un nouveau séparateur, et obtention d'une nouvelle marge.



**Approche multi-échelle:** la population de travail est renouvelée à chaque itération du processus d'apprentissage, et on note **3 phases**.

**Apprentissage****Position**

La **population de travail** est une distribution uniforme de taille  $N_1$  dans une hypersphère de rayon  $\beta_1$ .  
Sélection de  $k_1$  points parmi une compression par clustering des points de la marge.

**Stabilisation**

La **population de travail** est une distribution uniforme de taille  $N_2 > N_1$  dans une hypersphère de rayon  $\beta_2 < \beta_1$ .  
Sélection de  $k_2$  points parmi une compression par clustering des points de la marge.

**Précision**

La **population de travail** est une normale de taille  $N_3$  correspondant à la distribution de Monte-Carlo.  
Sélection des  $k_3$  points les plus proches de la frontière définie par le séparateur.

**•Validation**

- Suivi de la valeur de l'état-limite
- Comparaison des marges successives
  - Suivi de la probabilité de défaillance
- Calcul FORM sur la surface SVM

**Objectif :** obtenir une surface proche de l'état-limite par l'évaluation d'un minimum de points d'apprentissage en procédant par étapes : d'un positionnement grossier à un affinage de la position dans les zones à forte densité de probabilité défaillante.

**Calcul**

La probabilité de défaillance est évaluée par Monte-Carlo sur la frontière analytique obtenue.  
Le nombre de simulations dépend du niveau de fiabilité cible.

**•Validation**

- Coefficient de variation
- Bootstrapping

Méthode SMART - Bilan.

### L'idée

Soit  $F$  le domaine de défaillance. Considérons une séquence de domaines de taille décroissante et inclus l'un dans l'autre :  $F_1 \supset F_2 \supset \dots \supset F_m = F$ . En fiabilité des structures, la défaillance est atteinte lorsque l'évaluation de la fonction de performance  $G$  est négative. La séquence des domaines peut alors être définie de la manière suivante :  $F_i = \{G(x) < y_i\}$ , où les  $y_i$  sont des valeurs décroissantes de la fonction de performance et  $y_m = 0$ . Cette décomposition en domaines est illustrée sur la figure III.15.

Utilisons la définition des probabilités conditionnelles pour écrire la grandeur recherchée :

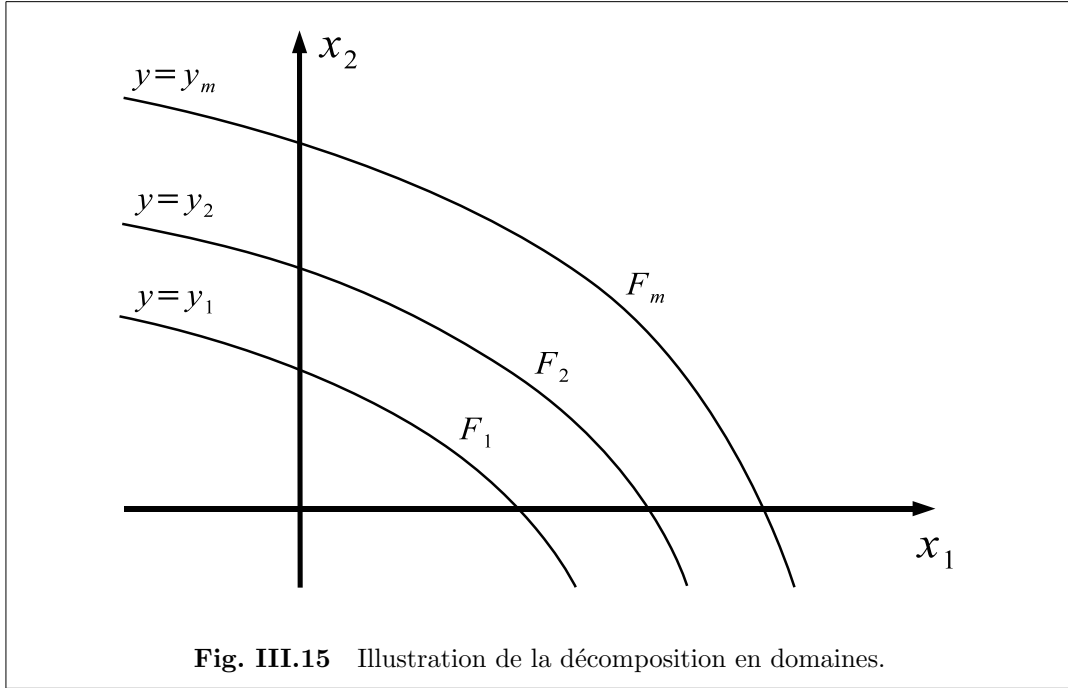
$$P_f = P(F) = P(F | F_{m-1})P(F_{m-1}) = \dots = P(F_1) \prod_{i=2}^m P(F_i | F_{i-1}) \quad (\text{III.2})$$

La probabilité de défaillance est exprimée par le produit de  $m$  probabilités conditionnelles. L'idée des *subsets simulations* est d'évaluer  $P_f$  par l'estimation des différents facteurs. De cette façon, même si  $P_f$  est très faible, en choisissant judicieusement la valeur des seuils intermédiaires, les différents facteurs à évaluer peuvent être assez grands et ainsi évalués de manière efficace par simulations. La règle utilisée pour le choix judicieux du seuil est la suivante : les seuils successifs sont choisis tels que les probabilités conditionnelles évaluées sont de l'ordre de 0,1. Pour évaluer  $P_f$ , il faut évaluer les probabilités :  $P(F_1)$  et  $\{P(F_{i+1} | F_i), i = 1, \dots, m-1\}$ . Pour la première probabilité, l'évaluation par Monte-Carlo classique est possible. Les probabilités conditionnelles nécessitent de générer une simulation conditionnée. L'idée est alors d'utiliser une méthode de chaînes de Markov basée sur l'algorithme de Metropolis Hastings, pour générer les échantillons conditionnés.

### Chaînes de Markov via l'algorithme de Metropolis Hastings

L'algorithme de Metropolis est une technique efficace pour simuler des échantillons ayant une densité arbitraire, voir Metropolis et Hastings [MRR<sup>+</sup>53, Has70]. On trouvera dans une référence française à ce sujet de C.P.Robert [Rob96], des détails sur les chaînes de Markov et notamment des critères de convergence de ce type d'algorithme.

Le principe de l'algorithme de Metropolis-Hastings permet, à partir d'échantillons distribués selon la densité  $q(\cdot | F_i)$ , de générer aisément d'autres points selon la même densité  $q(\cdot | F_i)$ . Les premiers points sont appelés germes de la chaîne



**Fig. III.15** Illustration de la décomposition en domaines.

de Markov. Considérons que nous possédons un germe  $\theta_1$ . Il s'agit de générer une chaîne d'échantillons  $\theta_1, \theta_2, \dots$  de manière itérative en calculant  $\theta_{k+1}$  à partir de  $\theta_k = [\theta_k(1), \dots, \theta_k(n)]$ .

L'algorithme de Metropolis-Hastings utilisé diffère un peu de sa version originale. Nous présentons ici la technique utilisée par Au et Beck :

- Un candidat est généré : pour chaque composante  $j = 1, \dots, n$ ,  $\epsilon_j$  est simulée selon une loi de probabilité  $p_j(\cdot \mid \theta_k(j))$ . Le ratio  $r_j$  est défini comme suit :  $r_j = q_j(\epsilon_j)/q_j(\theta_k(j))$ . On fixe  $\tilde{\theta}(j) = \epsilon_j$  avec une probabilité  $\min\{1, r_j\}$  où  $\tilde{\theta}(j) = \theta(j)$  avec une probabilité  $1 - \min\{1, r_j\}$ .
- Point accepté ou refusé : la fonction de performance est évaluée pour  $\tilde{\theta}$ . Si ce nouveau point appartient au domaine  $F_i$  alors  $\theta_{k+1} = \tilde{\theta}$  ; sinon ce nouveau point est rejeté et  $\theta_{k+1} = \theta_k$ .

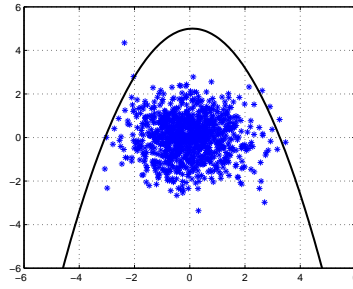
Le choix de la loi de probabilité  $p_j(\cdot \mid \theta_k(j))$  détermine la manière dont le point candidat est dévié du point dont il est issu. Les chaînes de Markov peuvent être considérées ici comme un chemin aléatoire dans l'espace à partir d'un point de départ : le germe. La loi de probabilité choisie est uniforme autour du point de départ. Il convient alors de fixer la largeur de l'intervalle. Un intervalle trop faible ne dévie pas le point de germe et l'espace exploré est trop proche. A l'inverse, un intervalle très large n'est pas bénéfique étant donné qu'il allonge le temps de la procédure, de

nombreux points étant rejetés. Une valeur comprise entre 1 et 3 apparaît satisfaisante. Une chaîne de Markov est réalisée à partir de chaque point de germe, ces points étant les réalisations générées à l'étape  $i$  et appartenant au domaine  $F_{i+1}$ . Ces chaînes vont définir la distribution conditionnée nécessaire pour évaluer la probabilité de l'étape  $i + 1$ .

### Subsets simulations - Algorithme de Au et Beck

Les étapes de l'algorithme sont décrites ici de manière succincte et permettent de comprendre l'enchaînement des opérations décrites dans les sections précédentes.

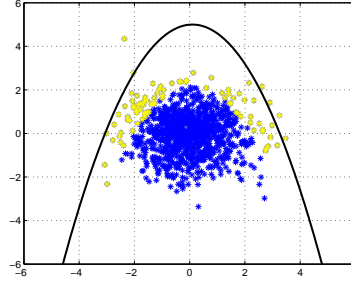
1. Transformation des variables de base dans l'espace standard.
2. Simulations de Monte-Carlo pour évaluer la première probabilité, taille  $N_{simu}$ .



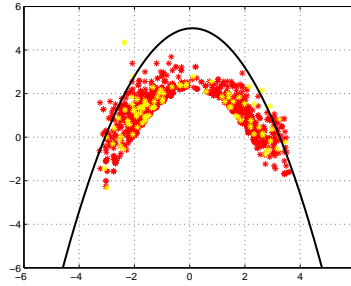
**Fig. III.16** Étape 2. La première étape consiste à évaluer  $P(F_1)$  par Monte-Carlo. Une simulation est ici réalisée dans l'espace standard.

3. Sélection d'une valeur  $y_1$  de l'état-limite telle que  $P(G \leq y_1) \approx 0, 1$ .
4. Sélection des points  $X$  tels que  $G(X) \leq y_1$ . Ces points sélectionnés vont servir de germe pour former des chaînes de Markov permettant de générer l'ensemble  $\{X \mid G(X) \leq y_1\}$ . La simulation conditionnelle obtenue figure III.18 a été générée par l'algorithme de Metropolis-Hastings à partir des germes.
5. Sélection d'une valeur  $y_2$  de l'état-limite telle que  $P(G \leq y_2 \mid G \leq y_1) \approx 0, 1$ .
6. Sélection des points  $X$  tels que  $G(X) < y_2$  et retour aux étapes précédentes tant que  $y_i > 0$ .

La probabilité de défaillance finale correspond au produit des probabilités obtenues à chaque étape.



**Fig. III.17** Étape 3. Une valeur de la fonction d'état-limite  $y_1$  est déterminée telle que  $P(G \leq y_1) \approx 0,1$ . Cette valeur constitue le premier seuil pour la méthode.



**Fig. III.18** Étape 4. Cette figure présente la simulation obtenue par l'algorithme de Metropolis. Cette simulation est distribuée selon la loi conditionnée par la valeur du premier seuil. Elle permet d'évaluer la probabilité  $P(F_2 | F_1)$ .

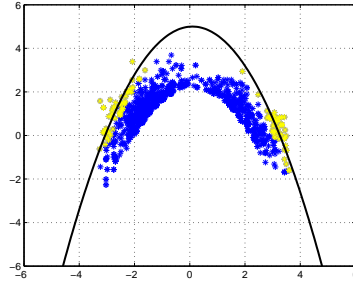
### Le contrôle du résultat

Le coefficient de variation de la probabilité de défaillance obtenue peut être évalué. De manière très succincte, ce coefficient de variation est estimé à partir des coefficients de variation intermédiaires qui sont pondérés par l'autocorrélation mesurée sur les chaînes de Markov. Sa formulation est la suivante [BA05] :

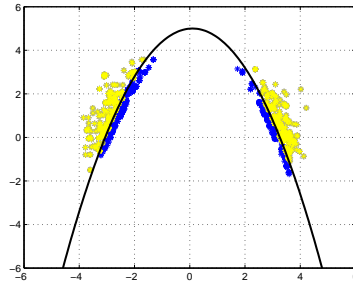
$$\delta^2 = \sum_{i=1}^m \delta_i^2 = \sum_{i=1}^m (1 + \gamma_i) \left( \frac{1 - p_{0i}}{p_{0i} N} \right) \quad (\text{III.3})$$

où  $\delta_i$  sont les coefficients de variation successifs,  $\gamma_i$  correspond à la corrélation des échantillons de deux pas de *subsets simulations* successifs,  $N$  le nombre de simulations nécessaires par pas de *subsets simulations*,  $p_{0i}$  la probabilité de défaillance conditionnée à chaque pas  $i$  et  $m$  le nombre de pas de *subsets simulations*.





**Fig. III.19** Étape 5. La valeur de  $G$  à 10% est déterminée par rapport à la simulation conditionnée obtenue.



**Fig. III.20** Étape 6. Le nouveau seuil est la valeur 0. On évalue sur la simulation conditionnée la probabilité de dépassement de ce dernier seuil. La probabilité de défaillance finale résulte du produit des probabilités obtenues jusque là.

### Le choix de 0,1

Une explication est donnée sur le choix de la probabilité cible pour le pas de *subsets simulations*. Considérons l'équation précédente III.3 avec  $\bar{\gamma}$  la moyenne des corrélations successives et  $p_0$  la valeur moyenne de la probabilité de chaque seuil, alors :

$$\delta^2 = m (1 + \bar{\gamma}) \left( \frac{1 - p_0}{p_0 N} \right) \quad (\text{III.4})$$

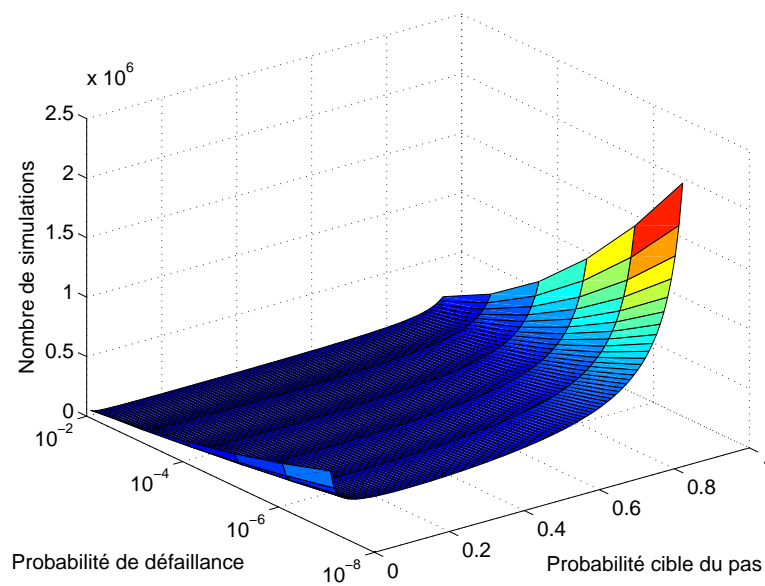
ainsi, il est possible d'obtenir le nombre total de simulations pour l'obtention de la probabilité de défaillance finale en fonction de la probabilité cible :

$$N_S = m N = m^2 (1 + \bar{\gamma}) \left( \frac{1 - p_0}{p_0 \delta^2} \right) \quad (\text{III.5})$$

D'après Au et Beck, la moyenne des corrélations est de l'ordre de 3. Fixons à 0,1 l'objectif de coefficient de variation final. Dans ce cas, la figure III.21 donne l'évolution

du nombre total de simulations en fonction de la probabilité de défaillance finale et de la probabilité cible choisie. L'ordre de grandeur à 10% de la probabilité cible par pas est justifié par le graphique.

Cependant, cette illustration ne prend pas en compte l'effet du choix de la probabilité cible du pas sur la corrélation des populations entre les pas. Si la probabilité cible du pas était plus grande (par exemple 30%), alors la population de germes serait plus importante (30% de la population totale), et la génération de la population du pas suivant nécessiterait des chaînes de Markov moins longues pour retrouver une population de taille équivalente. Or, la convergence de l'algorithme de Metropolis-Hastings n'est atteinte qu'à partir d'un grand nombre de générations. Un mauvais choix de la probabilité cible aurait une mauvaise influence sur la convergence de la méthode.



**Fig. III.21** Évolution du nombre total de simulations de la méthode de simulations par *subsets* en fonction du choix de la probabilité cible du pas et pour différentes valeurs de la probabilité de défaillance. Ce graphique permet de justifier le choix de la probabilité cible à environ 10%.

### III.6.2 Algorithme SVM et subsets simulations - <sup>2</sup>SMART

#### Pourquoi une version d'apprentissage basée sur les subsets simulations ?

La procédure SMART, détaillée dans la première partie de ce chapitre est assez efficace mais, pour des probabilités de défaillance très faibles, la population nécessaire à l'évaluation par simulation de Monte-Carlo est trop importante. De plus, le plan initial généré pour la méthode SMART est essentiel pour une convergence robuste : aucune zone de défaillance ne doit être oubliée. Ce détail prend de l'importance en cas de dimension importante du problème et lorsque la probabilité est faible : l'espace est très grand et il est difficile d'obtenir rapidement une idée précise de la répartition des zones de défaillance.

La méthode par *subsets simulations*, seules, offre une solution efficace pour résoudre ce type de problème mais cela reste une méthode de simulation. Pour un coefficient de variation du résultat suffisant, un grand nombre de simulations est encore nécessaire.

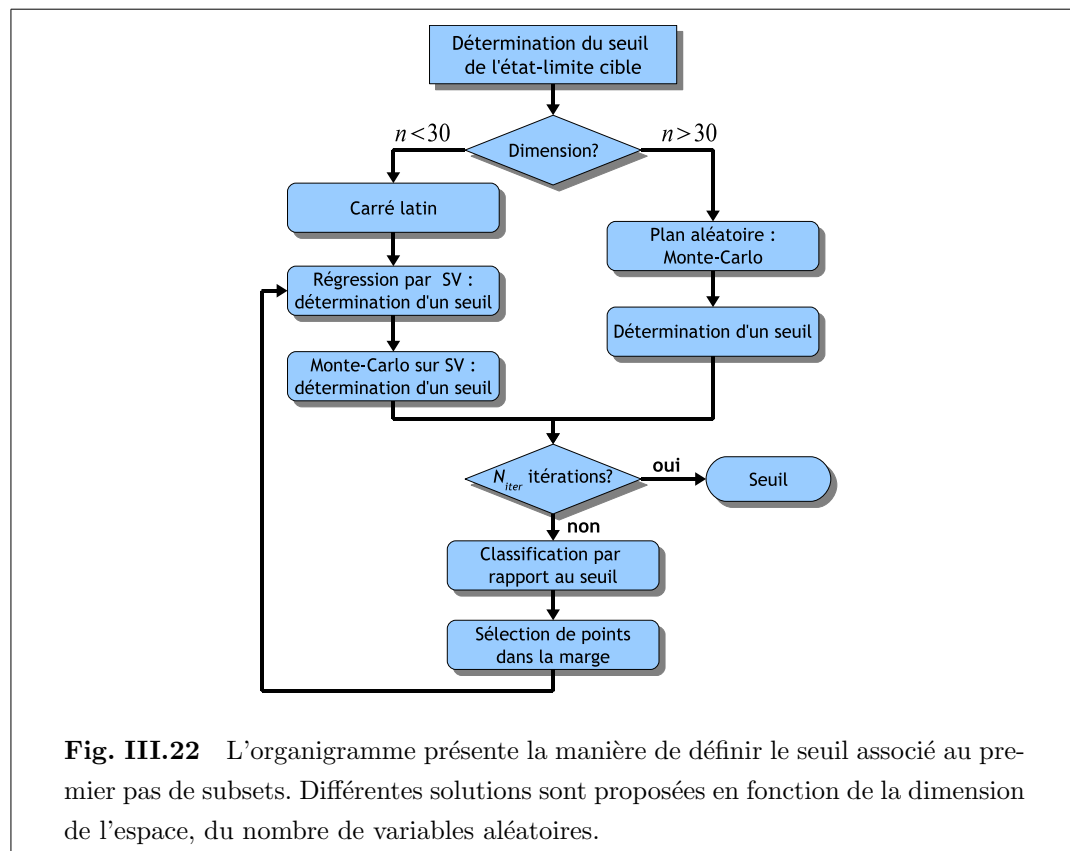
Nommée <sup>2</sup>SMART, pour *Subsets by Support-vector Margin Algorithm for Reliability esTimation*, l'approche couplée SVM et *subsets simulations* est présentée dans cette section. Il s'agit de cumuler les points forts des deux méthodes. L'approche par *subsets simulations* permet le calcul d'une faible probabilité de défaillance, en amenant l'algorithme dans les zones importantes en terme de densité de probabilité. Les SVM permettent de rendre efficaces les méthodes de Monte-Carlo : diminuer le coefficient de variation de l'estimation de la probabilité de défaillance à nombre de simulations imposé, ou réduire le nombre de simulations requis pour atteindre un coefficient de variation de l'estimation de la probabilité de défaillance imposé. Dans le cadre des *subsets*, les SVM vont permettre d'expliciter des frontières entre sûreté et défaillance, mais uniquement dans les zones d'intérêt pour la mesure de probabilité et ce à chaque pas de *subsets*. L'exercice de construction de la frontière est simplifié puisque les zones de construction sont restreintes.

En couplant SVM et *subsets*, nous proposons de résoudre de manière efficace un problème de fiabilité. L'approche par classification de la procédure de simulation, effectuée à chaque pas de *subsets* permet le calcul à coût raisonnable de la probabilité de défaillance par rapport à chaque seuil. Cette approche couplée SVM et *subsets* a fait l'objet d'une publication [DL07].

### Le premier pas de subset

La première étape consiste à évaluer une probabilité de  $\alpha\%$  par une simulation de Monte-Carlo. Une population de  $N$  individus est générée par Monte-Carlo. Comme il est décrit dans la section III.4, plusieurs populations de travail sont employées pour construire un substitut de l'état-limite, puis les  $N$  échantillons sont évalués sur la formulation analytique obtenue. L'algorithme SMART est donc utilisé dans cette première étape avec tout de même une particularité. Il est nécessaire de définir le seuil de la fonction d'état-limite garantissant la probabilité de défaillance de  $\alpha\%$

**(a) Détermination du seuil.** Avant d'utiliser l'algorithme SMART pour la construction de l'état-limite, il s'agit de définir le seuil de la fonction d'état-limite autour duquel notre méthode doit être appliquée. En fonction de la dimension de l'espace, deux solutions sont proposées pour évaluer ce seuil. Ces solutions sont illustrées sur la figure III.22.



- Dans le cas de problèmes de taille courante (de 2 à 20-30 variables environ), un premier plan par carré latin, plan de taille  $k$ , est effectué. Ce type de plan d'expériences est choisi pour des raisons similaires à celles évoquées dans la section III.4.2. C'est-à-dire : plan aléatoire et optimisation de la dispersion des premiers points. Sur ces  $k$  premiers points d'étude, une régression par la technique des SVM est réalisée. Les  $N$  individus générés par Monte-Carlo sont évalués sur cette surface analytique. Il est possible de déterminer une valeur de l'état-limite à  $\alpha\%$ .

Afin d'affiner la valeur de seuil obtenue, et ainsi s'assurer de travailler à une valeur de seuil proche du  $\alpha$  optimal, cf. section III.6.1, la qualité de la régression autour de la zone du seuil, la zone stratégique, est réalisée. Il s'agit d'ajouter plusieurs points d'apprentissage autour du seuil puis de répéter l'opération de calcul du seuil. Pour sélectionner des points proches du seuil en question, la méthode utilisée pour SMART est appliquée. Par un calcul de classification, une marge autour du seuil est définie. Les nouveaux points d'apprentissage sont choisis parmi les points de la marge. De cette façon, une meilleure précision de la régression autour du seuil est obtenue, et donc également une valeur plus précise pour ce seuil.

- pour un espace de dimension plus importante (supérieure à 20-30 variables), le premier plan d'expériences est une simulation de Monte-Carlo d'environ 100 individus. Après l'évaluation de ces points, une approximation est fournie pour ce seuil à  $\alpha\%$ , la précision réalisée étant généralement acceptable.

**(b) Évaluation de la probabilité de défaillance.** Connaissant à présent la valeur du seuil, il est alors possible d'effectuer la méthode SMART pour l'optimisation d'une surface de SVM par classification selon les deux classes séparées par le seuil défini. La stratégie décrite pour SMART est donc employée : approche multi-échelle, choix optimal des points d'apprentissage successifs : clusters des points de la marge et sélection des points instables, optimisation des paramètres des SVM par la validation croisée...

A la fin de ce premier processus d'apprentissage, la valeur de la première probabilité de défaillance est calculée par l'évaluation des  $N$  échantillons de Monte-Carlo sur la formulation analytique de la limite.

### Les pas suivants

**(a) La technique d'apprentissage.** Comme pour les subsets, les points défaillants de l'étape  $i$  permettent de générer la distribution qui va permettre l'évaluation de la probabilité de défaillance à l'étape  $i + 1$ . Ainsi, les  $\alpha\%$  points défaillants parmi les  $N$  individus de l'étape  $i$  sont employés pour produire la distribution conditionnelle de l'étape suivante par l'algorithme de Metropolis-Hastings.

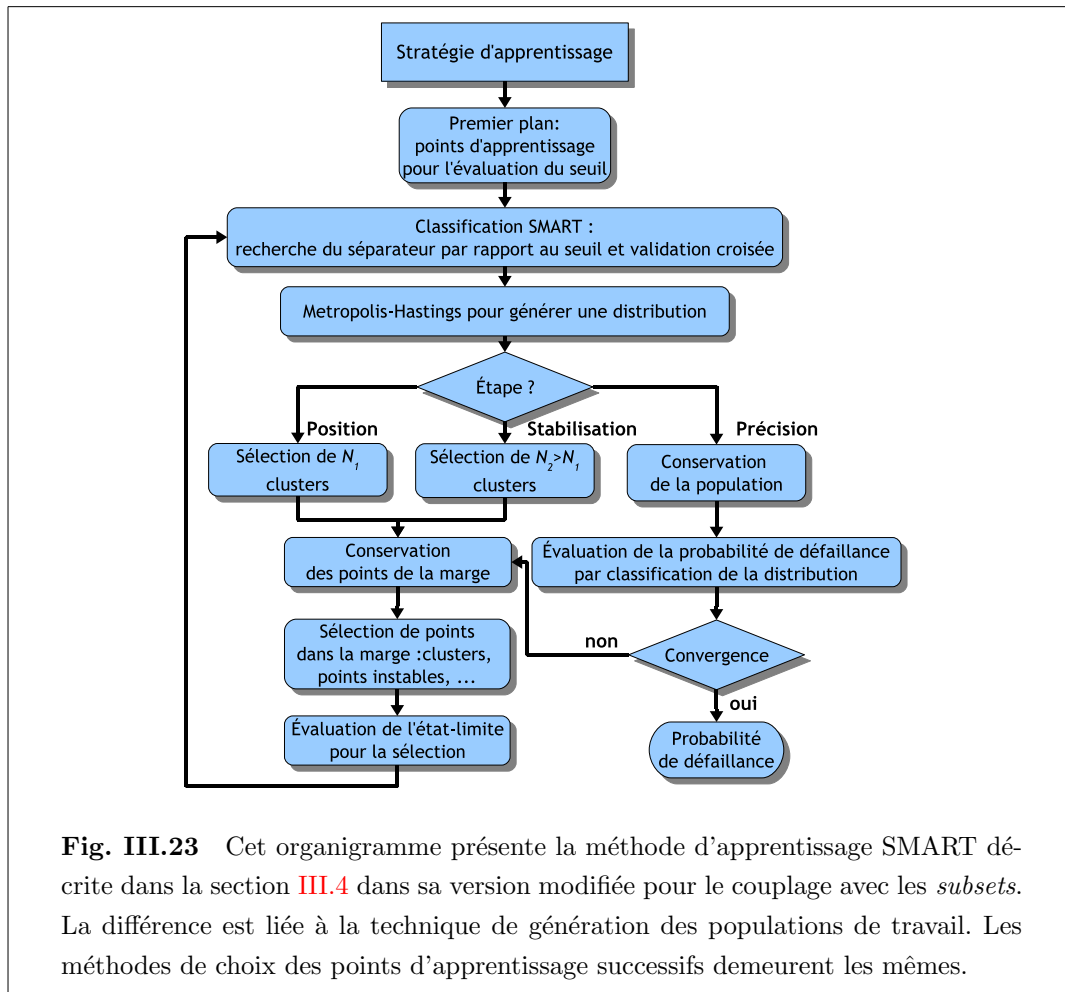
Cependant, notre méthode d'apprentissage de la limite nécessite de générer, pour chaque itération de l'apprentissage, une nouvelle population. En l'occurrence, cette population doit être conditionnée de la même manière que la population générée pour l'évaluation de la probabilité de défaillance. L'algorithme de Metropolis-Hastings est ainsi utilisé pour la construction des populations de travail pour chacune des étapes de la construction du séparateur.

De plus, pour conserver l'approche multi-échelle de l'apprentissage, la dispersion de ces populations de travail dépend de chacune des étapes de l'apprentissage : positionnement, stabilisation, précision. Pour permettre d'utiliser ce concept, nous jouons sur le ratio de l'acceptation de point  $r_j$  par l'intermédiaire du coefficient de variation de la densité  $q_j$ , voir algorithme Metropolis-Hastings section III.6.1. Ce ratio normalement fixé à 1, puisque les variables considérées sont ici dans l'espace standard, le fait de le fixer à 2 augmente la probabilité qu'un point plus éloigné soit accepté. Cela permet d'obtenir une dispersion de la chaîne de Markov plus importante.

De cette manière :

- pour l'étape de positionnement : une population de  $N_1$  individus est générée à l'aide de l'algorithme de Metropolis-Hastings piloté par un écart-type supérieur à 1. Ainsi les  $N_1$  individus sont dispersés sur le domaine conditionné par la limite précédente. Ce qui aboutit à des propriétés du plan équivalente au plan initial par carré latin dans le domaine conditionné.
- Pour l'étape de stabilisation : la méthode utilisée est identique, mais un nombre plus important d'individus est généré, soit  $N_2 > N_1$ .
- Pour l'étape de précision : une population de  $N$  individus est générée par la méthode originale de Metropolis-Hastings. Pour chaque itération de cette dernière phase, la population de travail est générée de manière identique à la population servant à l'évaluation de la probabilité de dépassement du seuil en cours.

Toutes les populations pour l'apprentissage du séparateur sont alors définies, il reste à déterminer le seuil pour exécuter l'optimisation du séparateur.



**(b) Détermination du seuil.** La technique est similaire à la détermination du premier seuil. La différence tient dans la méthode de sélection des premiers points d'apprentissage pour la régression. Les points supports du séparateur de l'étape précédente sont sélectionnés. Quelques points de la première population de travail sont ajoutés à la base d'apprentissage (clusters). Une régression est réalisée sur les points de cette base d'apprentissage.

En utilisant la population générée par Metropolis-Hastings pour évaluer la probabilité de défaillance, il est possible d'obtenir le seuil à  $\alpha\%$ . De la même manière que pour le premier pas, la régression est affinée par quelques points d'apprentissage dans la zone proche du seuil.

Le schéma de l'apprentissage est résumé dans la figure III.23. L'algorithme est terminé lorsque le seuil atteint est 0. La probabilité de défaillance finale est calcu-

lée, comme dans les *subsets simulations*, par le produit de toutes les probabilités intermédiaires.

### Un premier exemple

	MC	FORM	SS(AB)	SS(AB)	<sup>2</sup> SMART
$N$ appels	$10^7$	66	5000	50000	500
$P_f$ ( $10^{-5}$ )	1, 60	1, 01	2, 51	1, 23	1, 60
$\beta$	4, 16	4, 26	4, 05	4, 22	4, 16
Cov	0, 25	-	0, 34	0, 12	0, 03

**Tab. III.4** Résultats du calcul de fiabilité sur le modèle du fil conducteur. La valeur de  $\gamma$  est fixée ici à 5.

Les étapes exposées sont illustrées sur l'exemple du fil conducteur. Des résultats obtenus par diverses méthodes de fiabilité sont présentés dans le tableau III.4. La méthode <sup>2</sup>SMART, approche couplée SVM et *subsets simulations*, donne un résultat proche de la solution exacte.

La figure III.24 montre le processus des *subsets simulations* par SVM. Les différentes zones où l'algorithme de classification SMART est employé se distinguent. Pour chaque valeur du seuil, les points d'apprentissage se regroupent dans les zones d'intérêt pour l'évaluation de la probabilité de dépassement de ces seuils successifs.

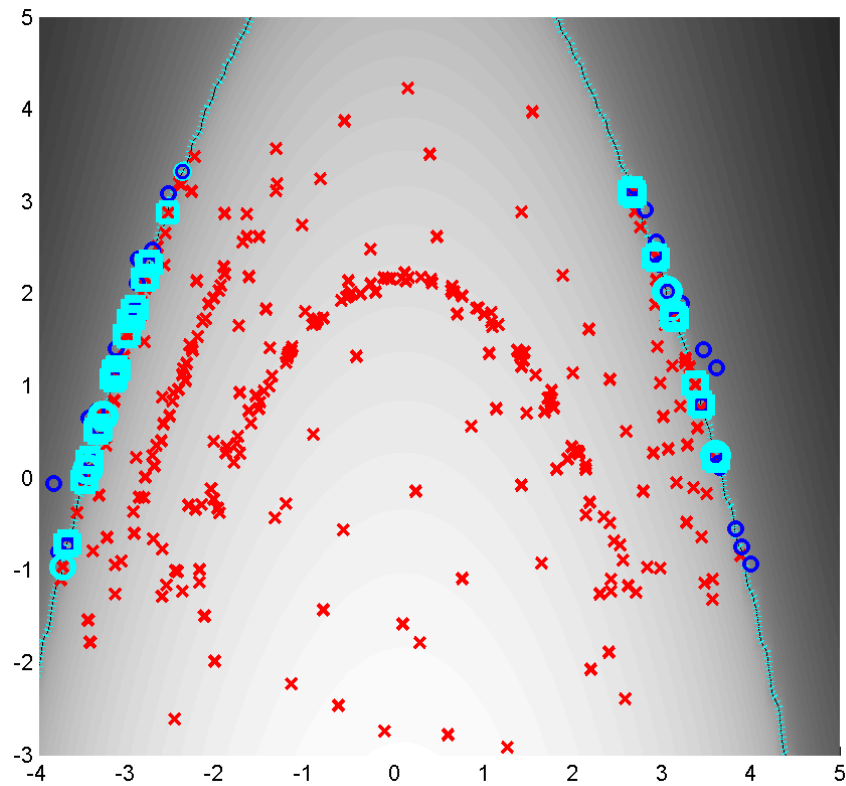
### Les points importants

De manière globale, les points importants sont équivalents à ceux de la technique d'apprentissage SMART, puisque la méthode est identique pour chaque pas de subsets à ce qui a été fait pour le couplage SVM et Monte-Carlo.

**(a) La régression.** Nous utilisons dans cet algorithme la régression par les SVM. Il est intéressant de normer la variable de sortie de la fonction d'état-limite ( $G$ ) pour plus de stabilité dans la méthode. La régression est réalisée sur la base d'apprentissage dans la première phase du processus de construction de la frontière pour chaque étape des *subsets*. La régression permet d'associer une valeur de la fonction d'état-limite au seuil cible (à 10%).

De plus, pour obtenir des points d'apprentissage dans une zone relativement proche de la frontière à affiner, le plan initial est réalisé en 2 temps :





**Fig. III.24** Séparateurs successifs obtenus pendant le processus de la méthode  $^2\text{SMART}$ . Cette image montre les points d'apprentissage successivement choisis par l'algorithme pour établir la limite entre les domaines de sûreté et de défaillance par rapport à chacun des seuils.

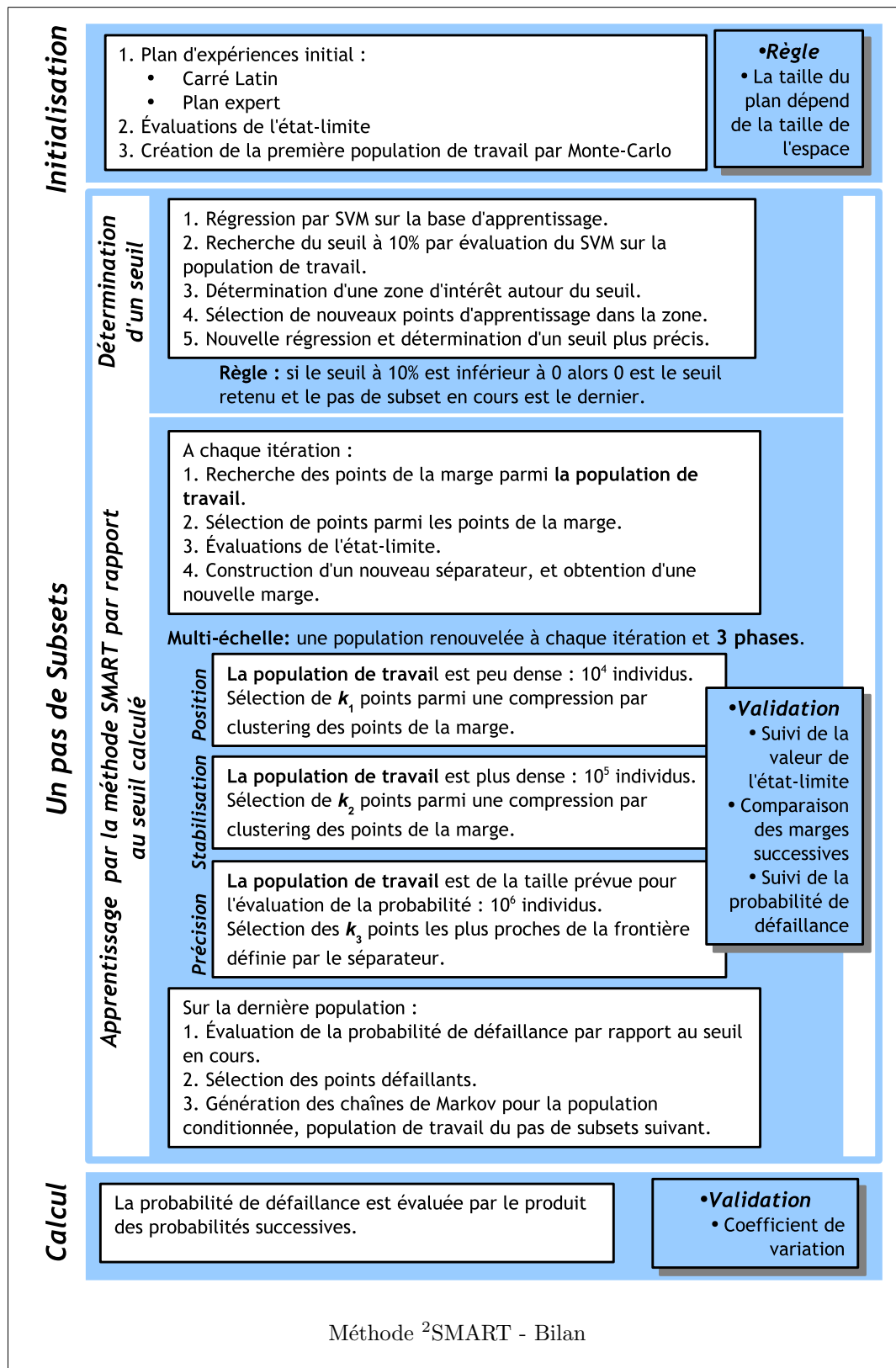
1. un premier temps consiste à réaliser un carré latin, de manière identique à l'algorithme d'apprentissage III.4.2 ;
2. à partir de la régression effectuée sur la base du plan initial, une zone d'intérêt est aisément définie. Quelques points sont ajoutés dans cette zone afin d'affiner la valeur de la régression dans la zone proche des 10%.

**(b) Le plan initial.** Le premier plan d'initialisation de l'algorithme est équivalent à l'algorithme d'apprentissage III.4.2. Les mêmes objectifs sont présents. La différence tient dans le fait que dans le cadre des *subsets*, nous ne sommes pas forcés d'obtenir des points dans le domaine de défaillance. La technique des *subsets* va permettre de diminuer les contraintes sur le plan initial.

(c) **La sélection de points dans la marge.** La méthode de sélection des points dans la marge est équivalente à ce qui a été décrit dans [III.4.3](#).

### Discussion

- Cette méthode couplée SVM et *subsets simulations* permet de conserver comme seul paramètre, la taille du premier plan d'expériences. Celui-ci dépend uniquement de la dimension de l'espace standard. Ceci est dû au fait que la probabilité recherchée est toujours la même : de l'ordre de 0,1.
- Nous aboutissons à rechercher la séparation des domaines uniquement dans les zones à forte probabilité de défaillance. En effet, l'algorithme de Metropolis-Hastings permet de générer les données que la classification doit séparer pour le pas de *subsets* suivant. Au fur et à mesure de la progression de l'algorithme, ces zones sont amenées vers les zones à forte densité de probabilité. L'exercice de classification est alors simplifié puisque la taille de l'espace à classer diminue avec les itérations.
- Cette méthode permet de chercher des probabilités très faibles : de l'ordre de  $10^{-3}$  jusqu'à  $10^{-6}$  voire  $10^{-9}$ .



## III.7 Conclusion

Après une présentation de la problématique de l'analyse mécano-fiabiliste, nous avons tenté de résoudre celle-ci par un algorithme basé sur les méthodes de l'apprentissage statistique, en particulier les SVM. L'algorithme de classification, dont la première version de J.E. Hurtado a été améliorée pas à pas, a abouti à la création de la méthode SMART. Le caractère global de la classification et la frontière explicite nous ont incité à continuer le travail de développement de cette approche. Différents critères ont été créés à partir des informations issues de la marge et de son évolution au cours des itérations. Les limites de Monte-Carlo ont nécessité l'orientation vers d'autres types de simulations et en particulier les *subsets simulations*. Le couplage entre classification et *subsets* a été simplifié par la maîtrise de l'algorithme de classification seul. Cette dernière technique permet d'aboutir à une méthode efficace pour obtenir la probabilité de défaillance, même très faible, d'un système complexe, que nous avons nommée <sup>2</sup>SMART.



## Chapitre IV

# Validation

Les méthodes développées dans le chapitre précédent sont testées sur des exemples courants de la littérature scientifique traitant des problèmes de fiabilité.

## Sommaire

---

<b>IV.1</b>	<b>Introduction</b>	<b>117</b>
<b>IV.2</b>	<b>Outils de développement</b>	<b>117</b>
<b>IV.3</b>	<b>Les critères de validation</b>	<b>117</b>
<b>IV.4</b>	<b>Description des exemples</b>	<b>118</b>
IV.4.1	Validation de l'apprentissage - la méthode SMART	118
IV.4.2	Validation de la méthode <sup>2</sup> SMART	120
<b>IV.5</b>	<b>Paramètres et suivi de la méthode d'apprentissage</b>	<b>122</b>
IV.5.1	Les paramètres de la méthode SMART	122
IV.5.2	Le suivi de la méthode SMART	123
IV.5.3	Discussion	129
<b>IV.6</b>	<b>Validation de la méthode <sup>2</sup>SMART</b>	<b>130</b>
IV.6.1	Analyse de formes complexes	130
IV.6.2	Précision et stabilité du résultat	131
IV.6.3	Influence de la dimension	132
<b>IV.7</b>	<b>Conclusion</b>	<b>133</b>

---

## IV.1 Introduction

L'OBJET de ce chapitre est la démonstration par l'exemple de l'intérêt de la méthode <sup>2</sup>SMART. C'est la seule façon de prouver son intérêt et sa validité. Les exemples sont issus de la littérature et permettent de tester les méthodes d'apprentissage sur des difficultés représentatives rencontrées en pratique dans les analyses de fiabilité.

Après la description des cas étudiés, les critères de validation et de suivi de l'algorithme sont présentés et illustrés.

## IV.2 Outils de développement

Les développements et expérimentations ont été réalisés sous MATLAB [MAT03]. Les méthodes SMART et <sup>2</sup>SMART, méthodes présentées dans le chapitre III, ont été intégrées à une version modifiée du logiciel FERUM [FER]. Cela a facilité l'utilisation des différentes lois de probabilité et des méthodes de transformation des variables dans l'espace standard. En ce qui concerne les techniques d'optimisation et d'utilisation des SVM, ainsi que des outils statistiques associés, la bibliothèque SPIDER [SPI06] a été utilisée.

## IV.3 Les critères de validation

Les méthodes d'apprentissage sont validées par rapport à différents critères :

- La valeur de la probabilité de défaillance : existe-t-il un biais sur la valeur de la probabilité par rapport à la valeur de référence calculée par simulation de Monte-Carlo ou par *subsets simulations* ?
- La dispersion de la valeur de la probabilité de défaillance obtenue autour de la valeur moyenne.
- le nombre d'évaluations nécessaires de l'état-limite à l'évaluation de la probabilité de défaillance. Ce critère étant la principale préoccupation de la thèse.



Finalement le critère de validation consiste en l'obtention d'un rapport nombre d'évaluations et précision satisfaisant, tout en se préservant de l'absence de biais dans le cas où l'on souhaite obtenir une précision supplémentaire en acceptant un surcoût de calcul.

## IV.4 Description des exemples

Plusieurs exemples sont issus de la thèse de P.H.Waarts [Waa00] qui rassemble différents cas tests de la littérature. D'autres proviennent du livre de M.Lemaire, [Lem08]. Les caractéristiques de ces exemples, parfois difficiles à appréhender, sont intéressantes pour mener la validation de nos méthodes :

- points de conception multiples ;
- surface d'état-limite bruitée ;
- unions et intersections d'états-limites ;
- faibles niveaux de probabilité ;
- courbure de l'espace ;
- influence de la dimension.

### IV.4.1 Validation de l'apprentissage - la méthode SMART

#### Le cas $R - S$

Le premier test de validation est naturellement le cas Résistance-Sollicitation. Les variables Résistance  $R$  et Sollicitation  $S$  sont définies dans le tableau IV.1.

Variabes aléatoires	Loi	Moyenne	Écart-type
$R$	Normale	7,0	1,0
$S$	Normale	2,0	1,0

**Tab. IV.1** [Cas  $R - S$ ] Définition des variables aléatoires pour le cas  $R - S$ .

### La surface bruitée

Le cas présenté ici est un état-limite bruité. La fonction à approximer est la suivante :

$$G(X) = X_1 + 2X_2 + 2X_3 + X_4 - 5X_5 - 5X_6 + 0,001 \sum_{i=1}^6 \sin(100X_i) \quad (\text{IV.1})$$

Les variables  $X_{1,\dots,6}$  sont définies dans le tableau [IV.2](#).

Variabes aléatoires	Loi	Moyenne	Écart-type
$X_{1,\dots,4}$	Lognormale	120	12
$X_5$	Lognormale	50	15
$X_6$	Lognormale	40	12

**Tab. IV.2** [Surface bruitée] Définition des variables aléatoires pour le cas de la surface bruitée.

L'intérêt de la fonction réside dans le bruitage de l'état limite. Les méthodes basées sur le gradient ne sont dans ce cas que peu, voire pas du tout, efficaces. Les résultats, tableau [IV.6](#), sont obtenus par simulations de Monte-Carlo sur le séparateur.

### Sphéroïde aplatie

La fonction à approximer est la suivante :

$$G(X) = 10 - \sum_{i=1}^{10} \frac{X_i^2}{1 + \frac{i}{10}} \quad (\text{IV.2})$$

Les variables sont définies dans le tableau [IV.3](#).

Variabes aléatoires	Loi	Moyenne	Écart-type
$X_{1,\dots,10}$	Normale	0	1

**Tab. IV.3** [Sphéroïde aplatie] Les variables aléatoires pour le cas de la sphéroïde.

### États-limites convexe et concave

L'état-limite convexe est défini par la formule suivante :

$$H(X) = 2,5 + 0,1(X_1 - X_2)^2 - \frac{X_1 + X_2}{\sqrt{2}} \quad (\text{IV.3})$$

L'état-limite concave est lui défini comme suit :

$$H(X) = 3 - 0,5(X_1 - X_2)^2 - \frac{X_1 + X_2}{\sqrt{2}} \quad (\text{IV.4})$$

Les variables  $X_1$  et  $X_2$  sont, dans les deux cas, deux variables gaussiennes normées et centrées. Ces deux fonctions d'état-limite présentent une forme particulière qui ne permet pas à l'approximation FORM une représentation satisfaisante du domaine de défaillance.

#### IV.4.2 Validation de la méthode <sup>2</sup>SMART

Les probabilités de défaillance des exemples précédents sont supérieures à  $10^{-4}$ . Cette valeur correspond à une valeur limite pour être traitée efficacement par la méthode SMART. Ces derniers ont permis de valider la procédure d'apprentissage de la surface d'état-limite par les SVM. Des exemples où la probabilité de défaillance est particulièrement faible sont choisis pour la validation de la méthode <sup>2</sup>SMART. Il s'agit de vérifier ici que le couplage SVM et méthode de simulation par *subsets* conserve une précision satisfaisante sur le résultat, malgré la multiplication des étapes et donc des approximations.

#### Une hyperbole

Ce premier exemple est intéressant par la forme de l'état-limite [?].

$$G(X) = \alpha(2X_1X_2 - \beta) \quad (\text{IV.5})$$

Les variables sont définies dans le tableau [IV.4](#).

Variabes aléatoires	Loi	Moyenne	Écart-type
$X_1$	Normale	3	1
$X_2$	Normale	5	2
$\alpha$	-	1	-
$\beta$	-	0,005	-

**Tab. IV.4** [L'hyperbole] Variables aléatoires pour le cas de l'hyperbole.

### Un problème système en dimension 2

Un autre exemple en 2 dimensions permet de vérifier le positionnement des points d'apprentissage au cours de l'algorithme. Un problème système est ici choisi car ce type de problématique perturbe non seulement l'approche par approximation FORM, mais également les méthodes de substitution classiques du type surfaces de réponse polynomiale. La fonction d'état-limite est définie par la formule suivante :

$$G(X) = \min \begin{cases} 3 + 0,1 (X_1 - X_2)^2 - \frac{X_1 + X_2}{\sqrt{2}} \\ 3 + 0,1 (X_1 - X_2)^2 + \frac{X_1 + X_2}{\sqrt{2}} \\ X_1 - X_2 + 3,5 \sqrt{2} \\ X_2 - X_1 + 3,5 \sqrt{2} \end{cases} \quad (\text{IV.6})$$

Les variables  $x_{1,2}$  sont des variables normales de moyenne 0 et d'écart-type 1.

### Le cas d'un système

L'état-limite est défini par la fonction suivante :

$$G(X) = \max \begin{cases} 2,677 - X_1 - X_2 \\ 2,500 - X_2 - X_3 \\ 2,323 - X_3 - X_4 \\ 2,250 - X_4 - X_5 \end{cases} \quad (\text{IV.7})$$

Les variables aléatoires sont définies par des variables gaussiennes normées centrées.

### Influence de la dimension

Nous étudions ici le cas de Rackwitz [Rac01]. L'article présente les résultats obtenus pour différents nombres de variables aléatoires.

La dimension de l'espace est un facteur déterminant dans le choix de la stratégie de résolution d'un couplage mécano-fiabiliste. Les méthodes de surfaces de réponse sont inadaptées lorsque la dimension devient importante.

$$G(X) = (n + a\sqrt{n}) - \sum_{i=1}^n X_i \quad (\text{IV.8})$$

avec la valeur  $a = 3$ . Ce cas est étudié pour  $n = 40, 100$  et  $250$ .

## IV.5 Paramètres et suivi de la méthode d'apprentissage

Le travail de validation de la méthode SMART a permis d'étudier l'influence des paramètres de l'algorithme. Ce travail a abouti à la conclusion qu'une méthode plus souple du point de vue de la gestion de ces paramètres était nécessaire. Mais il a permis également de définir des indicateurs pour le suivi de l'algorithme.

### IV.5.1 Les paramètres de la méthode SMART

Les paramètres à fixer pour l'utilisation de la méthode SMART sont les suivants :

- l'étendue du tirage aléatoire uniforme ;
- le nombre de simulations de Monte-Carlo sur la surface SVM ;
- le nombre de points ajoutés à chaque itération ;
- les règles d'ajouts de points d'apprentissage : les points de la marge, les points les plus proches, les points changeant de classe.

Les deux premiers critères dépendant de la probabilité de défaillance finale, inconnue *a priori*, nous ont conduit dans le développement de la méthode <sup>2</sup>SMART qui fixe ces valeurs.

La méthode SMART a été validée sur des cas où la probabilité de défaillance ne descend pas sous la valeur de  $10^{-4}$ . Dans ce cas de figure, le premier plan est effectué dans une hypersphère de rayon 4 et le nombre de simulations de Monte-Carlo est fixé à  $10^6$ .

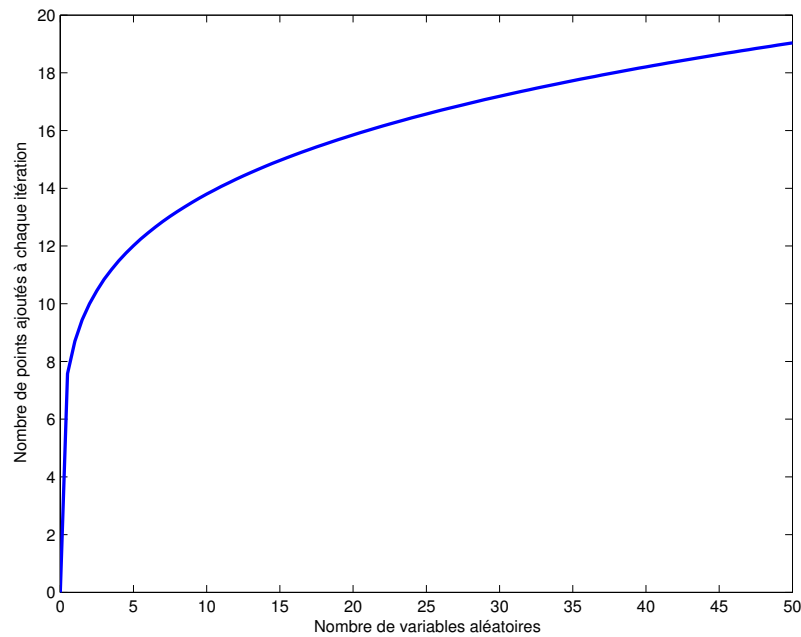
Les critères concernant le nombre de points ajoutés à chaque itération et leur répartition dans les différentes catégories dépendent de plusieurs paramètres :

- la taille de l'espace ;
- la précision que l'on souhaite obtenir.

Dans un souci de validation de la méthode et pour vérifier l'absence de biais, une solution plutôt robuste a été mise au point. Le nombre de points d'apprentissage ajoutés à chaque itération suit la courbe présentée en figure [IV.1](#).

Chacun de ces points est sélectionné en fonction de la phase d'apprentissage (position, stabilisation, précision, cf. [III.4.4](#)) dans les catégories suivantes (en proportion) :

- clusters des points de la marge ;
- points instables ;
- points les plus proches de la marge.



**Fig. IV.1** Évolution du nombre de points sélectionnés pour enrichir la base d'apprentissage en fonction de la dimension du problème.

Les proportions de points dans les différentes catégories sont de l'ordre de :

- $[100\% - 0\% - 0\%]$ , pour la phase de position ;
- $[40\% - 10\% - 50\%]$ , pour la phase de stabilisation ;
- $[0\% - 10\% - 90\%]$ , pour la phase de précision ;

Les choix qui ont été faits sont arbitraires. Il s'agit de valider en priorité la procédure et non pas le jeu de paramètres. Un mode plus dégradé permet également d'obtenir des résultats satisfaisants. L'expert peut selon ses objectifs adapter la procédure.

Ces paramètres ajustés à la méthode SMART ont été conservés pour la méthode <sup>2</sup>SMART qui les utilise à chaque pas de *subsets*.

#### IV.5.2 Le suivi de la méthode SMART

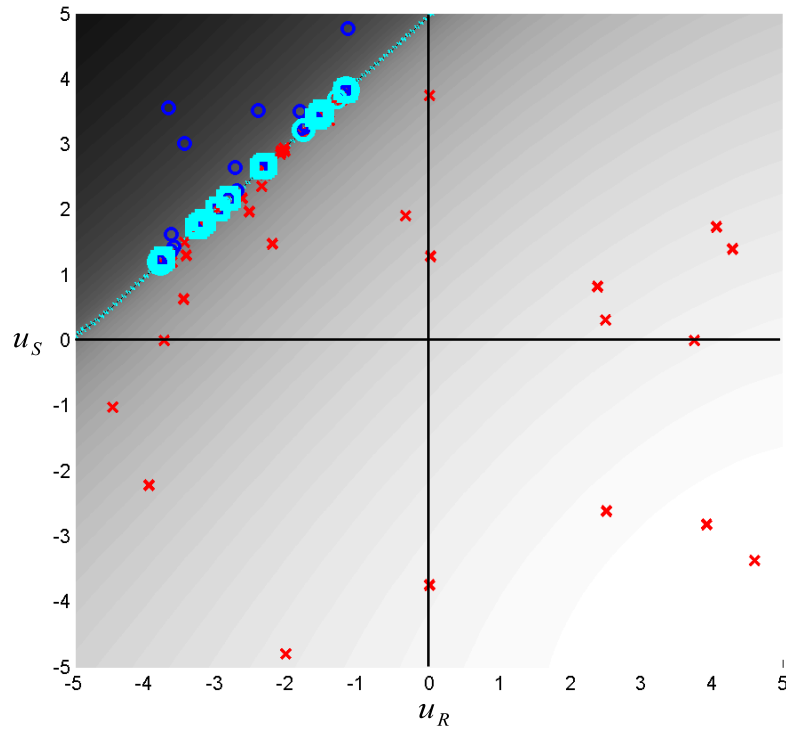
La validité de la valeur de la probabilité de défaillance obtenue par la méthode SMART est le principal intérêt. Cependant, afin de vérifier l'approximation locale obtenue par le séparateur autour des zones à forte densité de probabilité de défaillance, une analyse FORM est effectuée sur le séparateur analytique. La comparaison de l'indice de fiabilité FORM avec celui obtenu par couplage direct est intéressante du

point de vue de la finesse d'approximation dans ces zones importantes pour la mesure de la défaillance.

Pour contrôler la progression de l'algorithme, certains critères sur la marge ont été définis (cf. III.4.6).

La visualisation graphique de l'état-limite et des points d'apprentissage utilisés a également permis le suivi de l'apprentissage.

### FORM et Monte-Carlo



**Fig. IV.2** [Cas  $R - S$ ] Représentation graphique du séparateur obtenu dans l'espace standard,  $u_r$  et  $u_s$  étant les variables normées centrées associées à  $R$  et  $S$ . Les points d'apprentissage sont amenés naturellement autour de la frontière et en particulier dans la zone apportant la majeure part de probabilité de défaillance.

Le résultat de l'apprentissage pour le cas  $R - S$  est visible sur la figure IV.2. Les points d'apprentissage sont répartis sur la frontière entre sûreté et défaillance. Une concentration plus importante de ces points est visible autour du point de défaillance le plus probable.

Les résultats obtenus par l'algorithme de classification, présentés tableau IV.5, sont les résultats numériques de  $10^6$  simulations effectuées sur le substitut de l'état-limite obtenu à partir de 50 évaluations du modèle mécanique.

	Exact	Monte-Carlo	FORM	SMART	FORM sur SVM
$N$ appels	-	$10^6$	6	50	50
$P_f$	$2,0 \cdot 10^{-4}$	$2,2 \cdot 10^{-4}$	$2,0 \cdot 10^{-4}$	$2,2 \cdot 10^{-4}$	$2,0 \cdot 10^{-4}$
$\beta$	3,54	3,51	3,54	3,51	3,54
cv	-	0,08	-	0,08	-

**Tab. IV.5** [Cas  $R - S$ ] Résultats du calcul de la probabilité de défaillance pour le cas  $R - S$  à l'aide de différentes méthodes.

Le résultat SVM est plus proche du résultat de Monte-Carlo que du résultat exact, tout simplement puisque la méthode d'évaluation est la même dans les deux cas. Le coefficient de variation du résultat en terme de simulation est correct, même si l'approximation obtenue est éloignée du résultat exact.

Il est intéressant de noter que dans ce cas précis, l'approximation FORM sur le substitut donne le résultat exact et égal au couplage FORM direct. L'approximation linéaire permet ici l'obtention du résultat analytique exact.

Le suivi de l'approximation FORM sur la surface SVM a montré également, sur le cas du fil conducteur (cf. III.3) une précision de l'approximation autour du point de défaillance le plus probable.

### Suivi de la probabilité et de la marge

Les cas de la surface bruitée et de la sphéroïde servent de support à la présentation de courbes obtenues lors du suivi de la convergence. La figure IV.3 présente l'évolution de la marge et la figure IV.4 présente la convergence de l'évaluation de  $P_f$  au cours du processus d'apprentissage.

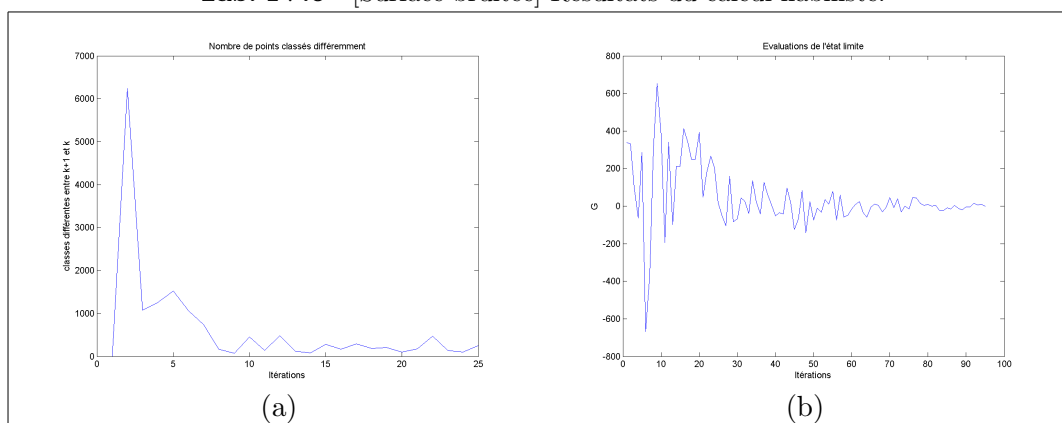
Le suivi de convergence sur le cas de la surface bruitée montre une instabilité de la convergence en  $P_f$  lorsqu'elle est mesurée par l'algorithme FORM sur le séparateur SVM, figure IV.4. Cela montre que la représentation de la surface est suffisamment précise pour reproduire le bruit et perturber ainsi l'algorithme FORM de la même manière que le couplage direct. En revanche, l'évaluation de  $P_f$  par Monte-Carlo



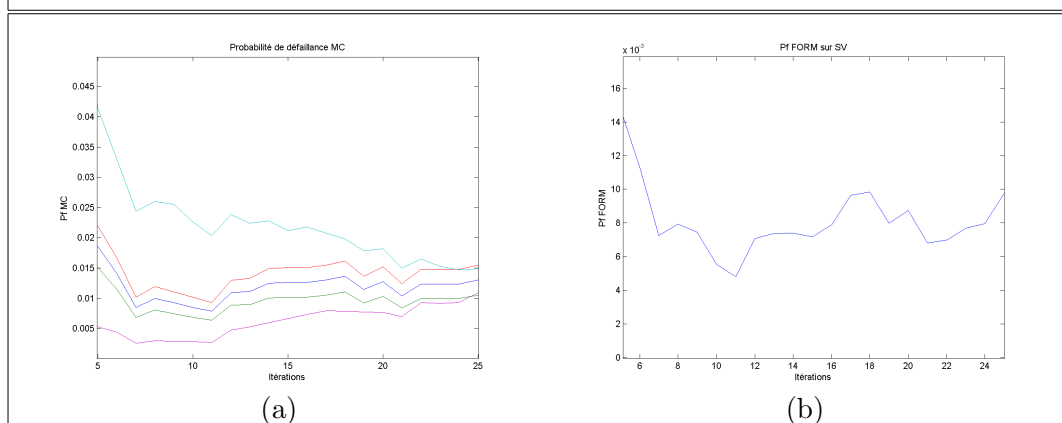
sur les marges inférieure et supérieure du séparateur est comprise dans l'intervalle de confiance déterminé par le nombre de simulations utilisées. L'incertitude sur la marge est inférieure à l'incertitude sur la valeur de la probabilité de défaillance.

	Exact	Monte-Carlo	FORM	SMART
$N$ appels	-	100000	44	50
$P_f$	$1,22 \cdot 10^{-2}$	$1,22 \cdot 10^{-2}$	$0,94 \cdot 10^{-2}$	$1,19 \cdot 10^{-2}$
$\beta$	2,25	2,25	2,35	2,26
cv	—	0,03	—	0,03

**Tab. IV.6** [Surface bruitée] Résultats du calcul fiabiliste.



**Fig. IV.3** [Surface bruitée] Suivi de convergence - Surface bruitée. (a) Points dont la classe évolue entre 2 itérations. (b) Évaluations de l'état-limite, la convergence vers 0 montre que les points d'apprentissage s'approchent de la frontière à chaque itération.

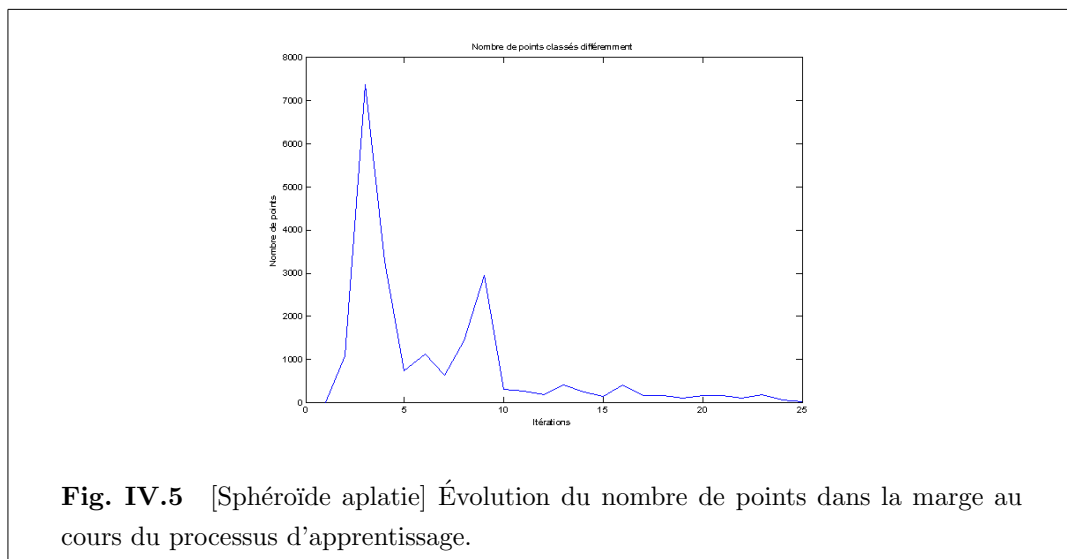


**Fig. IV.4** [Surface bruitée] Suivi de convergence - Surface bruitée. Convergence de l'évaluation de la probabilité de défaillance : Monte-Carlo (a) et FORM (b).

Le suivi du nombre de points de la marge permet également de vérifier l'évolution de l'apprentissage. La figure IV.5 montre que 100 points sont classés différemment entre les 2 dernières étapes du processus. Cela représente, sur  $10^5$  simulations, une différence de  $10^{-3}$  sur la probabilité de défaillance.

	Monte-Carlo	FORM	SVM
$N$ appels	-	204	50
$P_f$	0,135	$4,3 \cdot 10^{-3}$	0,141
$\beta$	1,10	3,33	1,08
cv	0,01	—	0,01

**Tab. IV.7** [Sphéroïde aplatie] Résultats du calcul fiabiliste pour le cas de la sphéroïde.



### Analyse graphique de la base d'apprentissage

L'analyse graphique de l'apprentissage, quand il est réalisable permet de contrôler le positionnement des points lors des itérations du processus d'apprentissage. Il permet également de voir la densité de points supports dans les zones à forte densité de probabilité de défaillance.

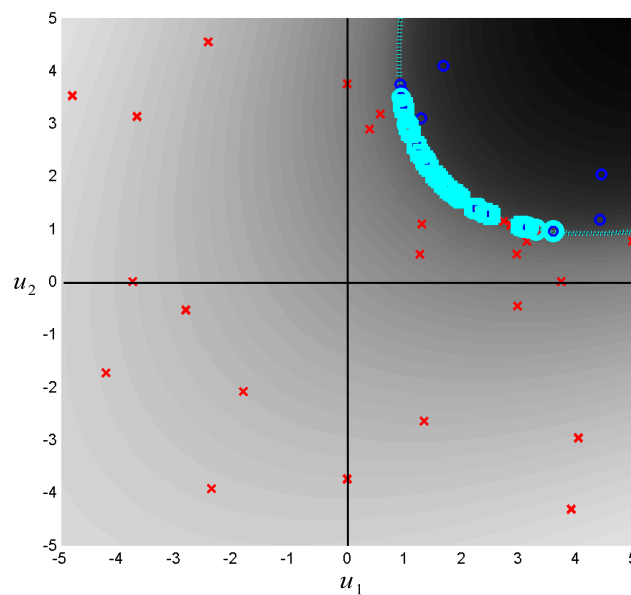
Le résultat de l'apprentissage pour les états-limites convexe et concave sont visibles sur les figures IV.6 et IV.7 respectivement. Les critères étudiés sont vérifiés sur ces deux cas : la dispersion du premier plan d'apprentissage et la concentration des points d'apprentissage dans les zones d'intérêt pour le calcul de la probabilité de défaillance.

	Exact	FORM	SMART
$N$ appels	-	8	50
$P_f$	$4,3 \cdot 10^{-3}$	$6,2 \cdot 10^{-3}$	$4,2 \cdot 10^{-3}$
$\beta$	2,63	2,50	2,63

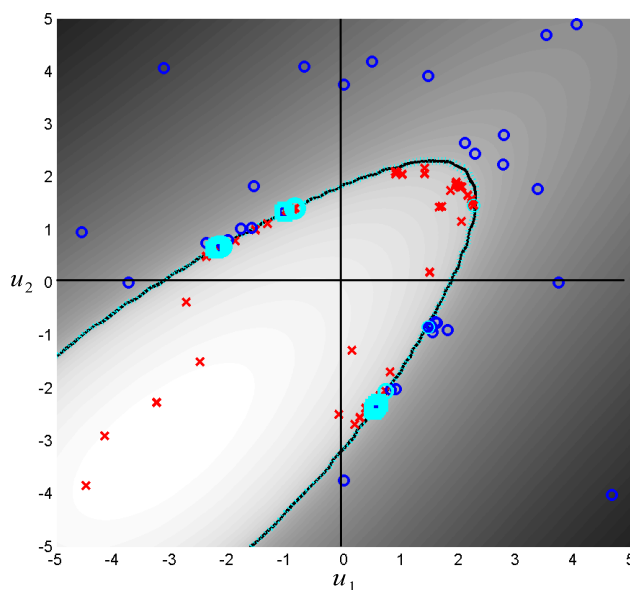
**Tab. IV.8** [Cas convexe] Résultats du calcul fiabiliste pour un état-limite convexe.

	Exact	FORM	SMART
$N$ appels	-	204	100
$P_f$	0,104	0,049	0,102
$\beta$	1,26	1,66	1,27

**Tab. IV.9** [Cas concave] Résultats du calcul fiabiliste pour l'état-limite concave.



**Fig. IV.6** [Cas convexe] Représentation graphique du séparateur obtenu pour l'état-limite convexe.



**Fig. IV.7** [Cas concave] Représentation graphique du séparateur obtenu pour l'état-limite concave.

### IV.5.3 Discussion

Les différents cas traités prouvent une certaine robustesse de l'apprentissage face aux différentes difficultés des états-limites testés. Les résultats obtenus par simulations sur les frontières analytiques sont proches des résultats obtenus par simulations de Monte-Carlo.

L'apprentissage réalisé par la méthode SMART aboutit à la construction d'une base de données dont les points se situent le long de la frontière séparant les deux domaines. Une concentration de points apparaît autour des points de défaillance. Ces caractéristiques voulues du plan d'expériences itératif, qui constituent le point fort de cette méthode d'apprentissage, permettent d'obtenir une méthode robuste pour l'évaluation de la probabilité de défaillance.

Cependant, certains paramètres nécessitent un ajustement en fonction de la probabilité de défaillance cible : par exemple le nombre de simulations de Monte-Carlo ou la dispersion du plan initial. Pour ces raisons, nous avons choisi d'adapter l'apprentissage de la frontière pour Monte-Carlo aux *subsets simulations*. La section suivante étudie la méthode <sup>2</sup>SMART sur des cas analytiques où cette approche est nécessaire. Nous notons dans les tableaux SS(AB) pour *subsets* par la méthode de Au et Beck.

## IV.6 Validation de la méthode <sup>2</sup>SMART

### IV.6.1 Analyse de formes complexes

Un point fort de l'utilisation des SVM pour la construction de l'état-limite tient dans la capacité à représenter des états-limites aux formes complexes.

La figure IV.8 présente le séparateur du pas final pour le cas de l'hyperbole (résultats tableau IV.10). Les différentes étapes sont marquées par la présence de nombreux points d'apprentissage.

	Exacte	FORM	SS(AB)	SS(AB)	<sup>2</sup> SMART
$N$ appels	-	66	3000	300000	250
$P_f$ ( $10^{-3}$ )	7, 20	6, 22	6, 50	7, 30	7, 40
$\beta$	2, 45	2, 50	2, 48	2, 44	2, 44
Cov	-	-	0, 22	0, 02	0, 01

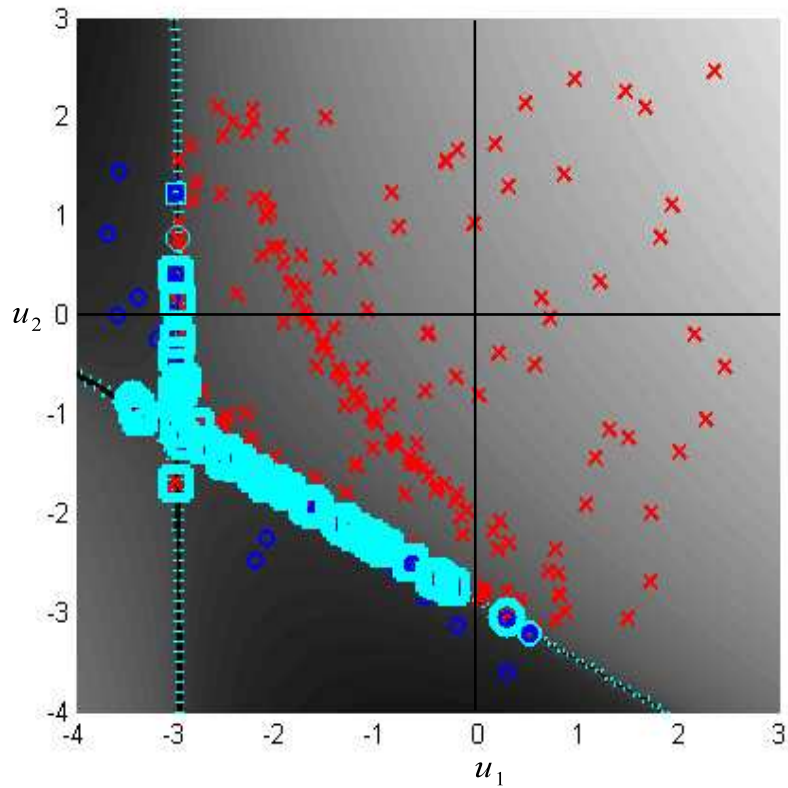
**Tab. IV.10** [L'hyperbole] Résultats du calcul fiabiliste pour le cas de l'hyperbole.

Cette capacité d'adaptation est d'autant plus remarquable sur des cas de fiabilité système. Ainsi les résultats du tableau IV.11 ont été obtenus par différentes méthodes de fiabilité sur un cas de fiabilité système à 5 variables, équation : IV.7. L'analyse de la figure IV.9 permet de juger de la capacité d'adaptation de la surface aux courbures imposées par l'état-limite système en 2D.

	Exacte	FORM	SS(AB)	SS(AB)	<sup>2</sup> SMART
$N$ appels	-	na	4000	40000	220
$P_f$ ( $10^{-4}$ )	2, 15	na	2, 60	2, 26	2, 38
$\beta$	3, 52	na	3, 47	3, 51	3, 49
Cov	-	na	0, 28	0, 09	0, 07

**Tab. IV.11** [Système 5 variables] Résultats du calcul fiabiliste pour le cas d'une fonction système.

Les méthodes d'approximation classiquement utilisées en fiabilité nécessitent une approche spécifique pour des cas systèmes tels que celui présenté ici. L'approximation FORM, dans son utilisation initiale, c'est-à-dire sans l'approche système, n'est ici pas applicable. Seules les méthodes de simulation permettent l'obtention de résultats corrects mais à un coût prohibitif. Notre approche est alors pertinente.



**Fig. IV.8** [L'hyperbole] Évolution des séparateurs au cours des itérations. La position des points d'apprentissage représente les pas des subsets.

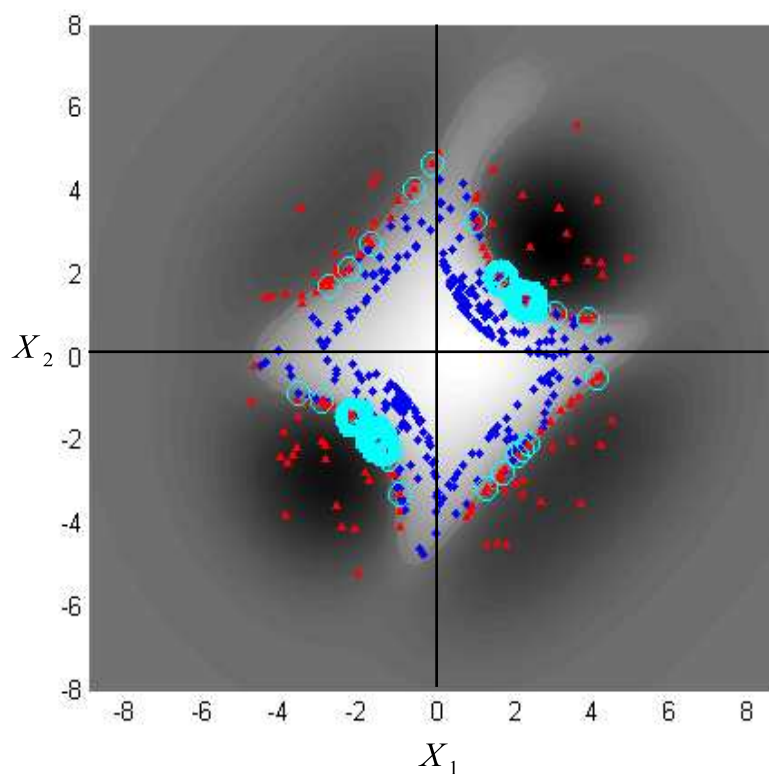
#### IV.6.2 Précision et stabilité du résultat

La méthode permet d'obtenir une précision satisfaisante pour un coût en terme de calcul qui l'est également. Mais au niveau de la précision, la valeur de la probabilité de défaillance est-elle affectée d'un biais ?

Les résultats présentés dans le tableau [IV.12](#) montrent une certaine efficacité de la méthode <sup>2</sup>SMART par rapport aux méthodes de simulation. Le coefficient de variation d'une valeur de 0,017 a été obtenu par 50 calculs différents pour les méthodes SS(AB) et <sup>2</sup>SMART. Le nombre de simulations nécessaires aux *subsets simulations* pour obtenir un coefficient de variation équivalent est 600 fois supérieur. Il est à noter que les valeurs de la probabilité de défaillance affichée pour ces méthodes est une valeur moyenne et donc que la méthode proposée n'a pas de biais sur cet exemple assez caractéristique.

	MC	FORM	SS(AB)	SS(AB)	<sup>2</sup> SMART
$N$ appels	$10^6$	12	3000	600000	1000
$P_f$ ( $10^{-3}$ )	2,16	1,34	2,35	2,22	2,21
$\beta$	2,86	3,00	2,83	2,84	2,84
Cov	0,02	—	0,23	0,017	0,017

**Tab. IV.12** [Système 2D] Résultats du calcul fiabiliste pour le cas système en dimension 2.



**Fig. IV.9** [Système 2D] Évolution des séparateurs au cours des itérations.

### IV.6.3 Influence de la dimension

L'efficacité de la méthode est jusqu'ici montrée sur des cas types. Mais il reste un point souvent problématique en fiabilité : l'influence de la dimension de l'espace sur la robustesse. En effet, la dimension de l'espace est un facteur déterminant dans le choix de la stratégie de résolution d'un couplage mécano-fiabiliste. Les méthodes

de surfaces de réponses sont généralement inadaptées lorsque la dimension devient importante.

Les résultats présentés dans le tableau IV.13 attestent de la tenue de la méthode à l'effet de la dimension (sur un cas test d'illustration).

L'erreur sur la valeur de la probabilité de défaillance ainsi que les coefficients de variation sont mesurés sur un échantillon de 20 résolutions.

On remarque l'absence de biais de la méthode <sup>2</sup>SMART par rapport aux valeurs de référence et le gain non négligeable d'appels à l'état-limite à coefficient de variation équivalent.

Dimension	40	100	250
$P_f$ ( $10^{-3}$ )	1, 98	1, 73	1, 59
SS(AB)	2, 4%	2, 3%	2, 4%
	$10^5$ appels/pas	$10^5$ appels/pas	$10^5$ appels/pas
SS(AB)	22%	17%	11%
	1243 appels/pas	2012 appels/pas	3569 appels/pas
<sup>2</sup> SMART	2, 8%	2, 2%	2, 6%
	1243 appels/pas	2012 appels/pas	3569 appels/pas
	0, 99	1, 01	1, 01

**Tab. IV.13** [Cas Rackwitz] Ce tableau présente les résultats obtenus par les *subsets simulations* et par <sup>2</sup>SMART sur le cas de Rackwitz pour 3 dimensions différentes de l'espace :40, 100 et 250. Pour les *subsets* nous affichons le coefficient de variation et le coût et pour <sup>2</sup>SMART nous ajoutons la moyenne de la probabilité de défaillance obtenue après 20 résolutions. Deux points importants : l'absence de biais (expérimental) et un gain soit en coefficient de variation à coûts équivalents ou un gain en coût à coefficients de variation équivalents.

## IV.7 Conclusion

La validation de l'apprentissage pour le calcul de la fiabilité a été menée dans ce chapitre. Les fonctions d'état-limite mathématiques ont été utilisées pour confronter l'apprentissage à différentes problématiques rencontrées en fiabilité des structures : problème de dimension, de points de conception multiples, de fonctions d'états-limites complexes, etc. La progression de l'apprentissage face aux difficultés est satisfaisante.



Des probabilités très faibles ont été calculées sans devoir ajuster le nombre de simulations de Monte-Carlo, ou encore la taille du plan initial. Les zones où se concentrent les points d'apprentissage correspondent aux zones où la densité pèse dans l'évaluation de la probabilité de défaillance, ce qui est pertinent pour économiser des appels à la fonction d'état-limite.

Le cumul de l'approximation à chaque pas de l'algorithme des *subsets* ne semble pas influencer de manière pénalisante sur la précision du résultat final, en effet ce cumul ne biaise pas le résultat en probabilité de défaillance. Les résultats obtenus sont donc satisfaisants. Les objectifs sont atteints.

Des exemples plus concrets, plus mécaniques, permettent de valider l'utilisation de la méthode <sup>2</sup>SMART pour une utilisation industrielle, dans le chapitre VI.

Le chapitre V, pour sa part, aborde la question de la régression et de l'analyse de sensibilité.

## Chapitre V

# Régression et analyse de sensibilité

Le méta-modèle SVM étant très souple et adaptable, la régression pour l'analyse de sensibilité devient intéressante à tester par cette voie.

## Sommaire

---

<b>V.1</b>	<b>Introduction</b>	<b>137</b>
<b>V.2</b>	<b>SVM et régression : quels apports pour la fiabilité ?</b>	<b>137</b>
V.2.1	Classification ou régression ?	137
V.2.2	Optimiser la régression par SVM	138
<b>V.3</b>	<b>Les produits de l'analyse de sensibilité</b>	<b>139</b>
V.3.1	Les moments statistiques	140
V.3.2	Quelques données théoriques sur les indices de Sobol'	140
V.3.3	L'évaluation des indices de Sobol'	144
V.3.4	Conclusion	147
<b>V.4</b>	<b>Estimation des moments statistiques et indices de Sobol'</b>	<b>147</b>
V.4.1	Méthode employée	147
V.4.2	Création de la base de données	148
V.4.3	Intervalle de confiance	148
V.4.4	La poutre en flexion	149
V.4.5	Treillis	151
V.4.6	Discussion	153
<b>V.5</b>	<b>Méthode Sparse Grid</b>	<b>155</b>
V.5.1	L'interpolation par la méthode de la grille	155
V.5.2	Optimisation de la grille	157
V.5.3	Premiers résultats	157
V.5.4	Discussion	158
<b>V.6</b>	<b>Conclusion</b>	<b>159</b>

---

## V.1 Introduction

**L**ES CHAPITRES précédents ont montré l'adéquation de l'outil SVM au calcul de fiabilité. L'emploi des SVM permet de traiter de manière tout à fait satisfaisante les problèmes à grands nombres de variables aléatoires, les états-limites de forme complexe, et les points de défaillance multiples notamment.

Ce chapitre vise à tester l'outil SVM dans un cadre autre que celui de la fiabilité : en analyse de sensibilité, autre volet de l'approche probabiliste en mécanique, afin de voir s'il est adapté à ce type d'analyse.

L'objectif est d'obtenir, à partir des points d'apprentissage générés par l'algorithme <sup>2</sup>SMART (ou d'un autre plan d'expériences), des informations sur les sensibilités de la fonction de performance par rapport aux variables d'entrée.

L'approche par classification pour la fiabilité a permis des avancées réelles sur deux points par rapport aux méthodes classiques :

- une souplesse de l'approximation en conservant une forte capacité de généralisation ;
- et surtout, une méthode de sélection optimale des points d'apprentissage par l'utilisation de la marge, qui plus est itérative et automatique.

Le premier item, souplesse de l'approximation, est transposable à l'analyse de sensibilité, puisque la propriété résulte de l'outil et du noyau utilisé. Le second objectif, sélection des points d'apprentissage, ne peut être satisfait par les SVM seuls. Un début de réponse est apporté en fin de chapitre par la présentation de l'approche par *sparse grid* et de quelques résultats.

## V.2 SVM et régression : quels apports pour la fiabilité ?

### V.2.1 Classification ou régression ?

Les deux applications, classification et régression, sont étudiées par la théorie de l'apprentissage statistique, et sont intéressantes pour une utilisation en mécanique probabiliste. La nuance est la suivante :

- en classification, à partir de quelques échantillons, un modèle mathématique permettant de reproduire la fonction  $\text{sign}(G(X))$  est recherché. En fiabilité, cela s'apparente à la recherche de l'état-limite :  $G(X) = 0$ . L'obtention d'un modèle mécanique de l'état-limite permet le calcul rapide de la probabilité de défaillance, mais également du (des) point(s) de conception.
- en régression, un modèle mathématique permettant de reproduire la fonction  $G(X)$  est recherché. Son obtention permet le calcul de la distribution de sa réponse, de sa moyenne et de ses moments statistiques d'ordres supérieurs.

Dans les deux cas, le modèle mathématique recherché correspond à une description globale du système. C'est-à-dire, une description sur l'ensemble du domaine de définition. Les deux activités sont intéressantes pour la fiabilité, mais la technique algorithmique utilisée dans les deux cas est différente, de même que les objectifs.

Il faut tout de même préciser qu'affecter une classe à un échantillon est plus simple que de mesurer l'interdépendance des variables.

### V.2.2 Optimiser la régression par SVM

Le point fort de la régression par SVM est contenu dans le principe SRM, cf. II.4.2. La complexité du méta-modèle est adaptée à la base d'apprentissage : l'optimisation de sa complexité est incluse dans le problème de régression SVM. Cependant, il reste une difficulté commune à toutes les procédures de régression : comment sélectionner les points d'apprentissage ?

L'objectif étant de connaître le comportement du modèle de manière générale, il nécessite un nombre de points idéalement répartis sur l'ensemble du domaine de définition. Il n'y a pas, pour le cas de la régression, de technique de sélection de points qui permette d'affiner le modèle obtenu. Le concept de marge des SVM ne prend un sens que dans le cadre de la classification. Tous les plans classiques sont par conséquent envisagés pour la régression.

Parallèlement au choix des points d'apprentissage, il est nécessaire de travailler sur l'optimisation des paramètres de la fonction noyau, critère important pour traiter le problème de la régression. Dans cette optique, différentes approches vont être couplées.

- Il est important de noter que la sélection des paramètres des fonctions noyaux ne connaît pas de règles précises. La méthode la plus présente dans la littérature consiste à tester différents couples de paramètres et à conserver le meilleur. Cette méthode est appelée "*cross validation*" et est à utiliser intelligemment

car il existe des risques de surapprentissage. En effet, le meilleur séparateur est sélectionné par rapport à la mesure de l'erreur empirique calculée sur les points d'apprentissage.

- Pour éviter ce surapprentissage dû à l'optimisation de la fonction sur la base d'apprentissage complète, il est préférable d'optimiser les paramètres par une mesure plus conforme à l'erreur de généralisation : la base des points est découpée en  $n$  groupes (de manière classique :  $n = 3$  ou  $5$ ). Sur  $n - 1$  groupes est réalisé l'apprentissage de la fonction et l'erreur est mesurée sur les points du dernier groupe. La comparaison des jeux de paramètres est alors faite sur la mesure d'une erreur de généralisation. Cette technique permet l'obtention d'une fonction à capacité de généralisation supérieure à celle qui serait obtenue par optimisation des paramètres sur la base complète.

La validation croisée associée à une approche de type rééchantillonnage, est utilisée dans ce chapitre pour optimiser la fonction noyau de type RBF.

### V.3 Les produits de l'analyse de sensibilité

L'analyse en sensibilité est également un volet important des études probabilistes. Il s'agit non plus de mesurer la probabilité d'un dépassement de seuil de la réponse, mais d'étudier globalement la réponse d'un modèle  $Y$ , modèle paramétré par un certain nombre  $n$  de variables aléatoires, représentées par le vecteur  $X$ .

$$\begin{aligned} f : \mathbb{R}^n &\rightarrow \mathbb{R} \\ X &\rightarrow Y = f(X) \end{aligned} \tag{V.1}$$

Deux aspects sont étudiés en analyse de sensibilité :

- la valeur centrale (typiquement la moyenne) et la dispersion de la réponse au travers de la mesure de ses moments statistiques ;
- l'influence de la variabilité des données d'entrée sur la variabilité de la donnée de sortie. Cette seconde étape permet de hiérarchiser l'influence des variables d'entrée sur la réponse d'un système.

Citons ici des références sur le traitement statistique de données : le livre de G.Saporta [Sap90], et un ouvrage en ligne mis à disposition par le NIST [NIS06].

### V.3.1 Les moments statistiques

L'obtention de la densité de probabilité de la réponse étant le plus souvent un objectif trop ambitieux, surtout lorsque le nombre de réalisations est restreint, l'étude de sensibilité s'intéresse généralement aux premiers moments statistiques de la réponse, *i.e.* moyenne, écart-type.

Soit  $X$  une variable aléatoire. On appelle moment d'ordre  $r$  de  $X$ , la quantité :

$$m_r(X) = E[X^r] \quad (\text{V.2})$$

Le moment d'ordre 1 est la moyenne.

On appelle moment centré d'ordre  $r$  de  $X$ , la quantité :

$$M_r(X) = E[(X - m_X)^r] \quad (\text{V.3})$$

Le moment centré d'ordre 2 est la variance, ceux d'ordre 3 et 4 sont respectivement associés à l'asymétrie et à l'aplatissement.

Ces différents moments peuvent être évalués par simulation directement sur le modèle mécanique. Pour des raisons évidentes de temps de calcul, notre objectif est de remplacer ce modèle par un modèle de substitution obtenu par régression sur une base d'apprentissage.

Différentes méthodes de régression sont comparées dans ce chapitre. Dans un souci de validation des méthodes de régression, les valeurs associées aux moments d'ordres 1 à 4 (moyenne, écart-type, asymétrie et aplatissement) sont évaluées par simulation sur modèle approché obtenu par le chaos polynomial ou par les SVM, puis comparées à leurs valeurs de références (obtenues par simulation directe).

### V.3.2 Quelques données théoriques sur les indices de Sobol'

#### Les indices de sensibilité

La variabilité de la réponse est une donnée intéressante pour la conception, mais connaître la ou les variables qui influent sur cette variabilité en est une bien plus utile. C'est ce que propose l'évaluation des sensibilités, qui étudie l'influence de perturbations des entrées sur la dispersion de la réponse. Elle permet la hiérarchisation des variables par rapport à leurs influences sur la réponse : influence seule ou couplée. La thèse de Julien Jacques présente un travail très intéressant sur les indices de sensibilité généralisés [Jac05]. Les généralités sur la sensibilité et les indices de

Sobol' sont issus de ses travaux basés sur les références principales suivantes : Saltelli et Sobol' [SCS00, Sob93].

En analyse de fiabilité, les indices de sensibilité associés à l'approximation FORM sont répandus. Les produits de l'analyse de fiabilité sont présentés par M.Lemaire dans [Lem05]. Ces indices mesurent l'influence des variables d'entrée sur l'indice de fiabilité. Cette analyse permet de hiérarchiser les variables par rapport à leur influence sur la défaillance. Il est également possible de quantifier l'influence des paramètres de la loi de probabilité des variables, ou de la corrélation, sur la probabilité de défaillance. Ces indices sont cependant associés à un point de défaillance. Il est donc nécessaire d'obtenir ce(s) point(s) par l'optimisation (voir section I.5).

Plus récemment, en mécanique probabiliste, les méthodes de sensibilité associées à la variance de la réponse ont été introduites, avec notamment les indices de Sobol', travail initié chez EdF par B.Sudret et repris par G.Blatman [Sud07, Bla06]. On s'intéresse, dans ce cas, à déterminer quelle part de la variance est due à la variance des variables d'entrée ou des groupes de variables d'entrée. Les indices de Sobol' sont basés sur la décomposition de la variance du modèle, décomposition unique [Sob93].

### Indice de Sobol' du premier ordre

Considérons la fonction suivante :

$$Y = f(X_1, \dots, X_n) \quad (\text{V.4})$$

Les variables  $X_i$  sont supposées indépendantes.

Pour mesurer l'importance de la variable  $X_i$  sur la variance de la sortie  $Y$ , est étudiée la décroissance de la variance de  $Y$  lorsque  $X_i$  est fixé à  $x_i^*$  :

$$V(Y \mid X_i = x_i^*) \quad (\text{V.5})$$

Cette dernière quantité est la variance conditionnelle de  $Y$  sachant que  $X_i = x_i^*$ . Cet indicateur est par conséquent dépendant du choix de la valeur de  $x_i^*$ . En considérant alors l'espérance de cette valeur pour tout le support de  $X_i$ , l'espace des possibles de  $x_i^*$ , le problème est résolu :

$$E[V(Y \mid X_i)] \quad (\text{V.6})$$

Plus cette valeur est petite, alors moins la variance de  $Y$  diminue lorsque  $X_i$  est fixée. Donc plus cette valeur est petite et plus la variable  $X_i$  est importante vis-à-vis de la variance de  $Y$ . En considérant le théorème de la variance totale, alors :

$$V(Y) = V(E[Y \mid X_i]) + E[V(Y \mid X_i)] \quad (\text{V.7})$$



pour un couple  $(X, Y)$  de variables aléatoires, où  $Y$  prend ses valeurs dans  $\mathbb{R}$  et  $X_i$  dans un ensemble fini ou dénombrable et sous la condition que la variance de  $Y$ ,  $V(Y)$ , est finie. Un nouvel indicateur de la sensibilité de  $Y$  à  $X_i$  est la variance de l'espérance de  $Y$ , conditionnellement à  $X_i$  :

$$V(E[Y | X_i]) \quad (\text{V.8})$$

Ainsi, plus cette grandeur est importante, plus la variable  $X_i$  est importante. Enfin, pour normaliser cet indicateur, l'indice de Sobol' du premier ordre est défini [Sob93], soit :

$$S_i = \frac{V(E[Y | X_i])}{V(Y)} \quad (\text{V.9})$$

Cet indice mesure la part de la variance de  $Y$  due à la variable  $X_i$ .

### Décomposition de la variance

La décomposition de Sobol' d'une fonction de variables à valeurs dans  $[0, 1]^n$ , s'écrit comme suit :

$$\begin{aligned} Y = f(X_1, \dots, X_n) = f_0 + \sum_{i=1}^n f_i(X_i) + \sum_{1 \leq i < j \leq n} f_{ij}(X_i, X_j) \\ + \dots + f_{1, \dots, n}(X_1, \dots, X_n) \end{aligned} \quad (\text{V.10})$$

où  $f_0$  est une constante et où les fonctions de la décomposition vérifient :

$$\begin{aligned} \int_0^1 f_{i_1, \dots, i_s}(X_{i_1}, \dots, X_{i_s}) dx_{i_k} = 0, \\ \forall k = 1, \dots, s, \quad \forall \{i_1, \dots, i_s\} \subseteq \{1, \dots, n\} \end{aligned} \quad (\text{V.11})$$

Si  $f$  est intégrable sur  $[0, 1]^n$ , alors elle admet cette décomposition qui est unique.

Il est possible d'interpréter les termes de la décomposition en prenant l'espérance de  $Y$ , puis l'espérance de  $Y$  conditionnellement à  $X_i$ , puis par rapport à  $X_i$  et  $X_j$ , etc. Il résulte alors :

$$\begin{aligned} f_i(X_i) &= E[Y | X_i] - f_0 \\ f_{i,j}(X_i, X_j) &= E[Y | X_i, X_j] - f_i(X_i) - f_j(X_j) - f_0 \\ f_{i,j,k}(X_i, X_j, X_k) &= E[Y | X_i, X_j, X_k] - f_{i,j}(X_i, X_j) - f_{i,k}(X_i, X_k) \\ &\quad - f_{j,k}(X_j, X_k) - f_i(X_i) - f_j(X_j) - f_k(X_k) - f_0 \\ &\dots \end{aligned} \quad (\text{V.12})$$

Ce qui s'écrit également :

$$\begin{aligned}
f_0 &= E[Y] \\
f_i(X_i) &= E[Y | X_i] - E[Y] \\
f_{i,j}(X_i, X_j) &= E[Y | X_i, X_j] - E[Y | X_i] - E[Y | X_j] + E[Y] \\
f_{i,j,k}(X_i, X_j, X_k) &= E[Y | X_i, X_j, X_k] \\
&\quad - E[Y | X_i, X_j] - E[Y | X_i, X_k] - E[Y | X_j, X_k] \\
&\quad + E[Y | X_i] + E[Y | X_j] + E[Y | X_k] - E[Y] \\
&\quad \dots
\end{aligned} \tag{V.13}$$

La variance  $V$  de  $Y$  se décompose, d'après la décomposition de Sobol' de la variance, en :

$$V = \sum_{i=1}^n V_i + \sum_{1 \leq i < j \leq n} V_{ij} + \dots + V_{1,\dots,n} \tag{V.14}$$

où

$$\begin{aligned}
V_i &= V(E[Y | X_i]) \\
V_{i,j}(X_i, X_j) &= V(E[Y | X_i, X_j]) - V_i - V_j \\
V_{i,j,k}(X_i, X_j, X_k) &= V(E[Y | X_i, X_j, X_k]) - V_{ij} - V_{ik} - V_{jk} - V_i - V_j - V_k \\
&\quad \dots \\
V_{1,\dots,n} &= V - \sum_{i=1}^n V_i - \sum_{1 \leq i < j \leq n} V_{ij} - \dots - \sum_{1 \leq i_1 < \dots < i_{n-1} \leq n} V_{i_1,\dots,i_{n-1}}
\end{aligned} \tag{V.15}$$

### Les indices de Sobol'

On peut alors définir les indices de Sobol'.

– indices du premier ordre :

$$S_i = \frac{V_i}{V} = \frac{V(E[Y | X_i])}{V} \tag{V.16}$$

– indices du second ordre, mesurant la sensibilité de la variance de  $Y$  par rapport aux variables  $X_i$  et  $X_j$ , sensibilité non prise en compte dans les indices du premier ordre :

$$S_{ij} = \frac{V_{ij}}{V} \tag{V.17}$$

– indices du troisième ordre :

$$S_{ijk} = \frac{V_{ijk}}{V} \tag{V.18}$$

– etc, jusqu'à l'ordre  $n$ .

Cette décomposition est analogue à la décomposition ANOVA, [Wik07b], réalisée en analyse de variance.

La somme des indices de Sobol' est égale à 1. L'interprétation est simple : plus l'indice est proche de 1, plus la variable ou l'interaction de deux ou plusieurs variables est importante.

### Les indices totaux

Le nombre d'indices augmente très rapidement avec le nombre de variables. Homma et Saltelli [HS96] ont alors introduit les indices de sensibilité totaux. Ces indices regroupent le poids de chaque variable dans la variance, le poids de la variable seule ainsi que son poids dans les interactions avec les autres variables.

$$S_{T_i} = \sum_{k \# i} S_k \quad (\text{V.19})$$

où  $\#i$  représente les ensembles d'indices contenant  $i$ .

### V.3.3 L'évaluation des indices de Sobol'

B.Sudret [Sud07] propose d'évaluer les indices de Sobol' après décomposition de la réponse sur le chaos polynomial. En effet en présentant le lien mathématique existant entre les indices de Sobol' et les coefficients de la décomposition sur le chaos, les simulations ne sont plus nécessaires pour évaluer les indices de Sobol'. Une simple combinaison des coefficients du chaos permet d'obtenir les sensibilités. Cependant, cette technique suppose que l'on fixe le modèle de régression à une base polynomiale.

Notre objectif étant d'utiliser la régression par les SVM, il faut revenir à la méthode originale de la simulation pour évaluer les sensibilités, c'est ce qui est détaillé dans cette section.

### Estimation des indices par Monte-Carlo

L'estimation de l'espérance de  $Y$ ,  $E[Y] = f_0$ , et de sa variance  $V$ , à partir d'un échantillon de  $N$  réalisations des  $n$  variables d'entrée :  $(x_{k1}, \dots, x_{kn})_{k=1, \dots, N}$  est obtenue

nue par :

$$\begin{aligned}\tilde{f}_0 &= \frac{1}{N} \sum_{k=1}^N f(x_{k1}, \dots, x_{kn}) \\ \tilde{V} &= \frac{1}{N} \sum_{k=1}^N f^2(x_{k1}, \dots, x_{kn}) - \tilde{f}_0^2\end{aligned}\tag{V.20}$$

**(a) Indices du premier ordre.** Estimer un indice du premier ordre consiste à estimer la quantité suivante :

$$\begin{aligned}V_i &= V(E[Y \mid X_i]) \\ &= E[E[Y \mid X_i]^2] - E[E[Y \mid X_i]]^2 \\ &= U_i - E[Y]^2\end{aligned}\tag{V.21}$$

La variance de  $Y$  est estimée par la formule [V.20](#). Il reste alors à estimer la quantité  $U_i$ . Cette quantité correspond à l'espérance du carré de l'espérance de  $Y$  conditionné par  $X_i$ . Pour tenir compte du conditionnement, on fait varier toutes les variables sauf la variable  $X_i$ . Cette estimation nécessite donc deux échantillons de réalisations des variables d'entrée.

$$\begin{aligned}\tilde{U}_i &= \frac{1}{N} \sum_{k=1}^N f(x_{k1}^{(1)}, \dots, x_{k(i-1)}^{(1)}, x_{ki}^{(1)}, x_{k(i+1)}^{(1)}, \dots, x_{kn}^{(1)}) \\ &\quad \times f(x_{k1}^{(2)}, \dots, x_{k(i-1)}^{(2)}, x_{ki}^{(1)}, x_{k(i+1)}^{(2)}, \dots, x_{kn}^{(2)})\end{aligned}\tag{V.22}$$

Ainsi, les indices du premier ordre sont obtenus par :

$$\tilde{S}_i = \frac{\tilde{V}_i}{\tilde{V}} = \frac{\tilde{U}_i - \tilde{f}_0^2}{\tilde{V}}\tag{V.23}$$

**(b) Indices aux ordres supérieurs.** Pour les indices de second ordre  $S_{ij} = \frac{V_{ij}}{V}$  :

$$\begin{aligned}V_{ij} &= V(E[Y \mid X_i, X_j]) - V_i - V_j \\ &= U_{ij} - E[Y]^2 - V_i - V_j\end{aligned}\tag{V.24}$$

La quantité  $U_{ij} = E[E[Y \mid X_i, X_j]^2]$  en faisant varier toutes les variables sauf  $X_i$  et  $X_j$ , soit :

$$\begin{aligned}\tilde{U}_i &= \frac{1}{N} \sum_{k=1}^N f(x_{k1}^{(1)}, \dots, x_{k(i-1)}^{(1)}, x_{ki}^{(1)}, x_{k(i+1)}^{(1)}, \dots, x_{k(j-1)}^{(1)}, x_{kj}^{(1)}, x_{k(j+1)}^{(1)}, \dots, x_{kn}^{(1)}) \\ &\quad \times f(x_{k1}^{(2)}, \dots, x_{k(i-1)}^{(2)}, x_{ki}^{(1)}, x_{k(i+1)}^{(2)}, \dots, x_{k(j-1)}^{(2)}, x_{kj}^{(1)}, x_{k(j+1)}^{(2)}, \dots, x_{kn}^{(2)})\end{aligned}\tag{V.25}$$

Ainsi, les indices du second ordre sont obtenus par :

$$\tilde{S}_{ij} = \frac{\tilde{U}_{ij} - \tilde{f}_0^2 - \tilde{V}_i - \tilde{V}_j}{\tilde{V}} \quad (\text{V.26})$$

La même technique est utilisée pour obtenir les indices des ordres supérieurs. Étant donné le nombre de calculs nécessaires pour l'évaluation des indices de tous les ordres (l'évaluation des indices d'ordre  $p$  nécessitant le calcul de tous les indices d'ordre inférieur), on se contente souvent des indices du premier ordre ainsi que des indices totaux.

**(c) Indices totaux.** Les indices totaux peuvent être estimés plus simplement :

$$S_{T_i} = 1 - \frac{V(\mathbb{E}[Y \mid X_{\sim i}])}{V} = 1 - \frac{V_{\sim i}}{V} \quad (\text{V.27})$$

avec  $V_{\sim i}$  qui correspond à la variance de l'espérance de  $Y$ , conditionnellement à toutes les variables sauf  $X_i$ . Donc pour obtenir  $V_{\sim i} = U_{\sim i} - \mathbb{E}[Y]^2$ ,  $U_{\sim i}$  est calculé par :

$$\begin{aligned} U_{\sim i} = \frac{1}{N} \sum_{k=1}^N & f(x_{k1}^{(1)}, \dots, x_{k(i-1)}^{(1)}, x_{ki}^{(1)}, x_{k(i+1)}^{(1)}, \dots, x_{kn}^{(1)}) \\ & \times f(x_{k1}^{(1)}, \dots, x_{k(i-1)}^{(1)}, x_{ki}^{(2)}, x_{k(i+1)}^{(1)}, \dots, x_{kn}^{(1)}) \end{aligned} \quad (\text{V.28})$$

ainsi :

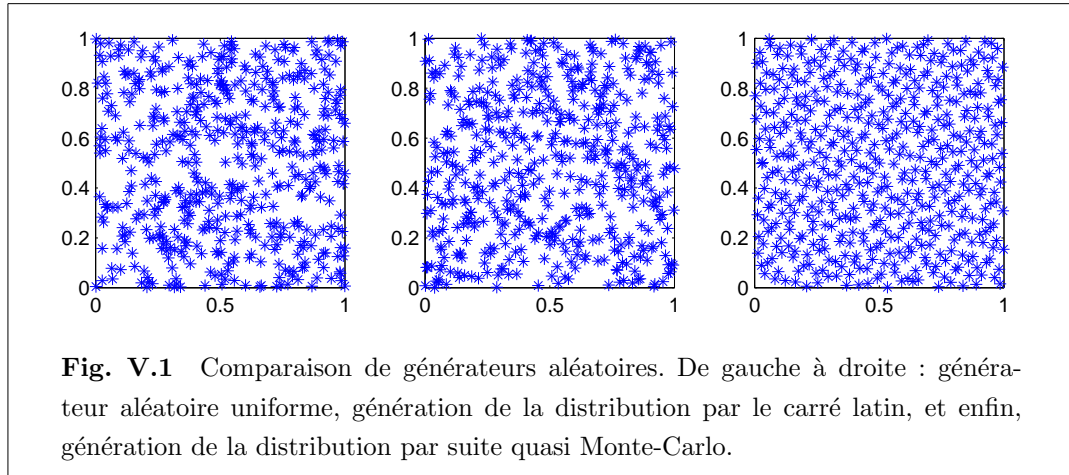
$$S_{T_i} = 1 - \frac{U_{\sim i} - \tilde{f}_0^2}{V} \quad (\text{V.29})$$

### Améliorations

Il est aisé de gagner en vitesse de convergence en utilisant un générateur quasi aléatoire en lieu et place de la simulation de Monte-Carlo. Il devient naturel alors pour l'évaluation des indices de Sobol' d'utiliser les suites quasi aléatoires du même auteur pour générer les réalisations. Il est intéressant d'accélérer la convergence de l'estimation des indices de Sobol', même si ces réalisations sont ensuite évaluées sur une formulation analytique de la fonction dans notre cas.

La figure [V.1](#) compare la répartition de 500 points issus de trois générateurs aléatoires ou quasi aléatoires. L'uniformité de la distribution obtenue est meilleure par suite quasi-aléatoire que par carré latin ou Monte-Carlo. Cette caractéristique explique le gain en convergence.

D'autres méthodes permettent l'estimation des indices de Sobol', comme la méthode FAST ou la méthode McKay. Plus d'informations sont disponibles dans la thèse de J.Jacques [[Jac05](#)].



**Fig. V.1** Comparaison de générateurs aléatoires. De gauche à droite : générateur aléatoire uniforme, génération de la distribution par le carré latin, et enfin, génération de la distribution par suite quasi Monte-Carlo.

### V.3.4 Conclusion

Cette section a présenté les produits de l'analyse de sensibilité et les méthodes permettant de les obtenir. Notre travail ne consiste pas en l'amélioration de ces techniques. Il s'agit de les mettre en œuvre sur le substitut de l'état-limite obtenu par régression.

Dans la suite nous mettons en œuvre sur des exemples simples l'estimation des produits de l'analyse de sensibilité pour comparer des méthodes de régression.

## V.4 Estimation des moments statistiques et indices de Sobol'

Pour présenter l'intérêt de la régression par SVM, quelques exemples sont ici présentés. Les produits de l'analyse de sensibilité sont calculés par simulations sur les modèles de substitution obtenus par la régression polynomiale ou la régression SVM. Les résultats sont comparés aux valeurs de référence pour ces quelques cas de validation. D'autres exemples sont également présentés dans le chapitre VI applications.

### V.4.1 Méthode employée

La théorie permettant d'utiliser la régression par les SVM a été détaillée dans le chapitre II. Il s'agit ici de présenter la démarche employée pour comparer les différentes méthodes de régression.

La régression par moindres carrés sur des polynômes de degrés 2 et 3 est comparée

à la régression par SVM utilisant une fonction noyau RBF dont le paramètre  $\sigma$  est optimisé par validation croisée. Le paramètre  $C$ , équilibrant erreur empirique et complexité du modèle, est fixé à la valeur  $10^4$  pour conserver une capacité de généralisation satisfaisante.

#### V.4.2 Création de la base de données

Pour différentes fonctions d'état-limite, une base de données est créée : évaluation de la fonction pour différents points déterminés par un plan d'expériences. Le plan d'expériences est un plan uniforme aléatoire par la technique du carré latin et généré dans le domaine de définition de la fonction. Ce type de plan est sélectionné, d'une part car il est habituellement choisi lors de régression sur des méta-modèles proches des fonctions noyaux et qu'il permet une vision globale à un coût minimal ; et d'autre part car ce plan a été choisi dans la méthode <sup>2</sup>SMART. De cette manière, il est possible de mener conjointement :

- la classification pour l'obtention de la probabilité de défaillance ;
- la régression sur le plan initial de la méthode <sup>2</sup>SMART pour l'obtention des indices de sensibilité généralisés, par exemple les indices de Sobol', par simulation sur la fonction de substitution obtenue.

Les résultats en indices de sensibilité (Sobol') sont obtenus par simulation quasi Monte-Carlo sur les modèles analytiques déterminés par la régression. Pour chaque cas, les 4 premiers moments statistiques sont également évalués par Monte-Carlo.

#### V.4.3 Intervalle de confiance

Pour les différentes techniques de régression (polynomiale et SVM), la dispersion des résultats est également mesurée. Une technique de *bootstrapping* est utilisée afin d'obtenir un intervalle de confiance sur les résultats statistiques.

Pour une taille de base d'apprentissage  $N$  donnée, 30 populations sont générées par tirages avec remise de  $N$  échantillons de la base de données principale. Les moments statistiques sont évalués par simulation sur le modèle de substitution obtenu par régression sur chacune des bases générées. Ainsi, pour chaque moment statistique, est obtenue une valeur moyenne du résultat ainsi que sa variance. Par la loi de Student est alors calculé un intervalle de confiance.

L'évolution de l'intervalle de confiance est analysée en fonction de la taille de la base de données choisie. Un resserrement des intervalles autour de la valeur réelle

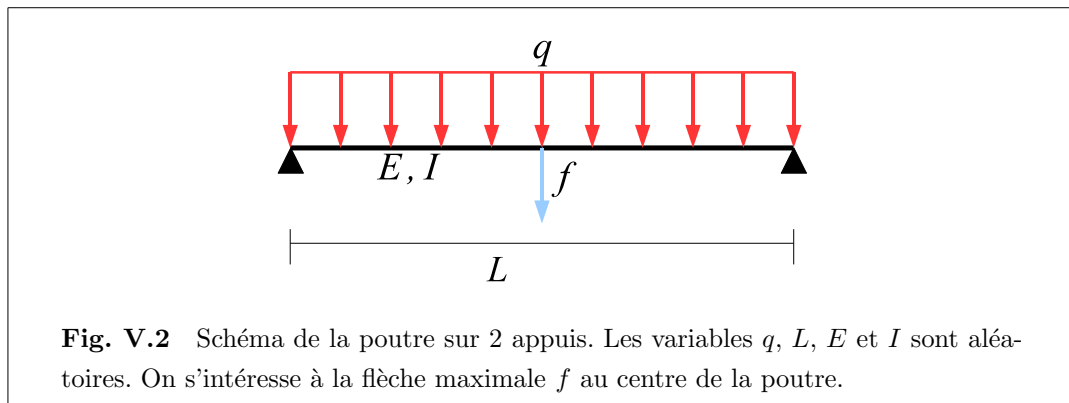
lorsque le nombre de points de la base augmente est naturellement constaté.

Cette technique a été utilisée afin de comparer le travail de régression par les SVM au rapport de fin d'étude de E. Noret [Nor06] réalisé autour de la régression sur le chaos polynomial. Certains exemples de ce rapport sont repris.

#### V.4.4 La poutre en flexion

##### Description du problème [Nor06]

Il s'agit d'une poutre sur 2 appuis, définie par sa longueur  $L$ , son module de Young  $E$  ainsi que son inertie  $I$ . La charge  $q$  est uniformément répartie sur la longueur de la poutre, figure V.2. Les variables sont décrites dans le tableau V.1.



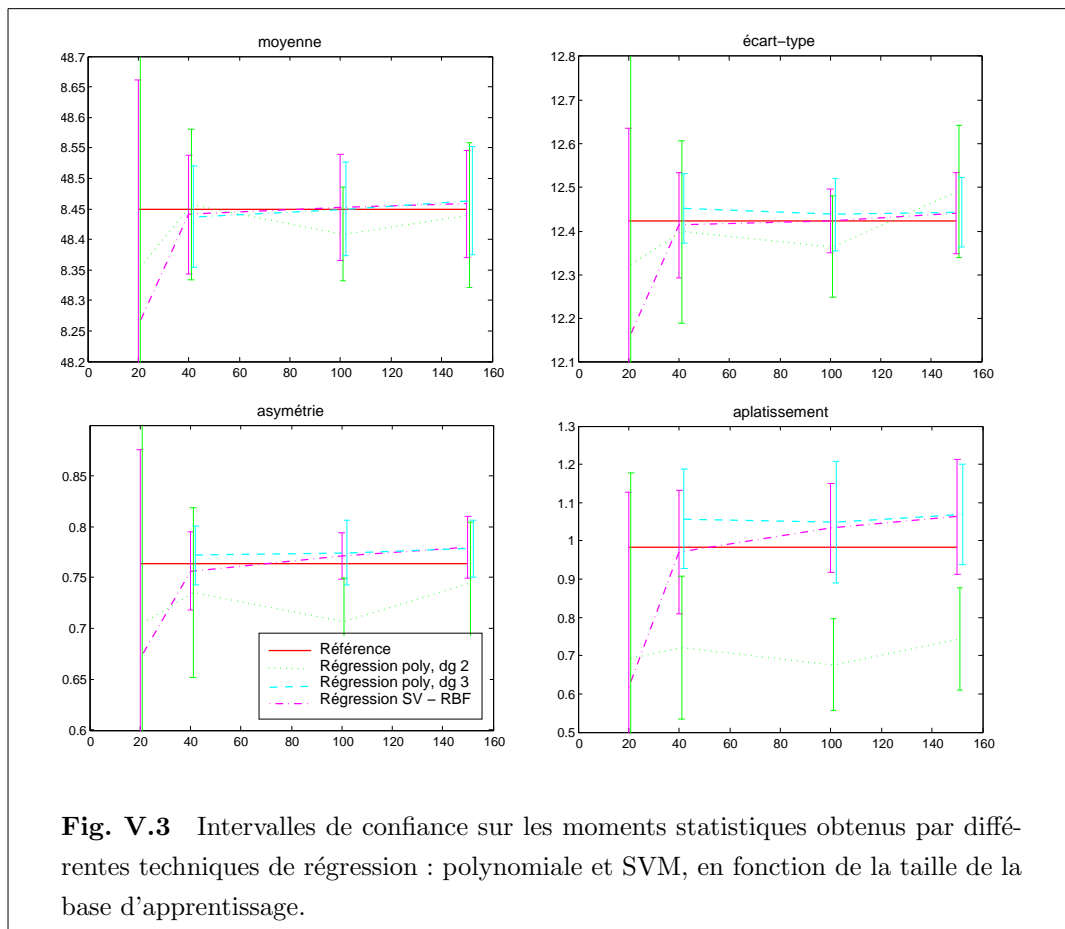
**Fig. V.2** Schéma de la poutre sur 2 appuis. Les variables  $q$ ,  $L$ ,  $E$  et  $I$  sont aléatoires. On s'intéresse à la flèche maximale  $f$  au centre de la poutre.

Variables aléatoires	Loi	Moyenne	Écart-type
$q$	Lognormale	5	1
$E$	Lognormale	210000	31500
$I$	Lognormale	$850 \cdot 10^4$	$425 \cdot 10^3$
$L$	Déterministe	6000	—

**Tab. V.1** Définition des variables aléatoires pour le cas de la poutre.

La flèche maximale  $f$  est évaluée par la formule :  $f = \frac{5}{384} \frac{qL^4}{EI}$ . L'intérêt de ce modèle est la possibilité du calcul exact des moments de la réponse.





**Fig. V.3** Intervalles de confiance sur les moments statistiques obtenus par différentes techniques de régression : polynomiale et SVM, en fonction de la taille de la base d'apprentissage.

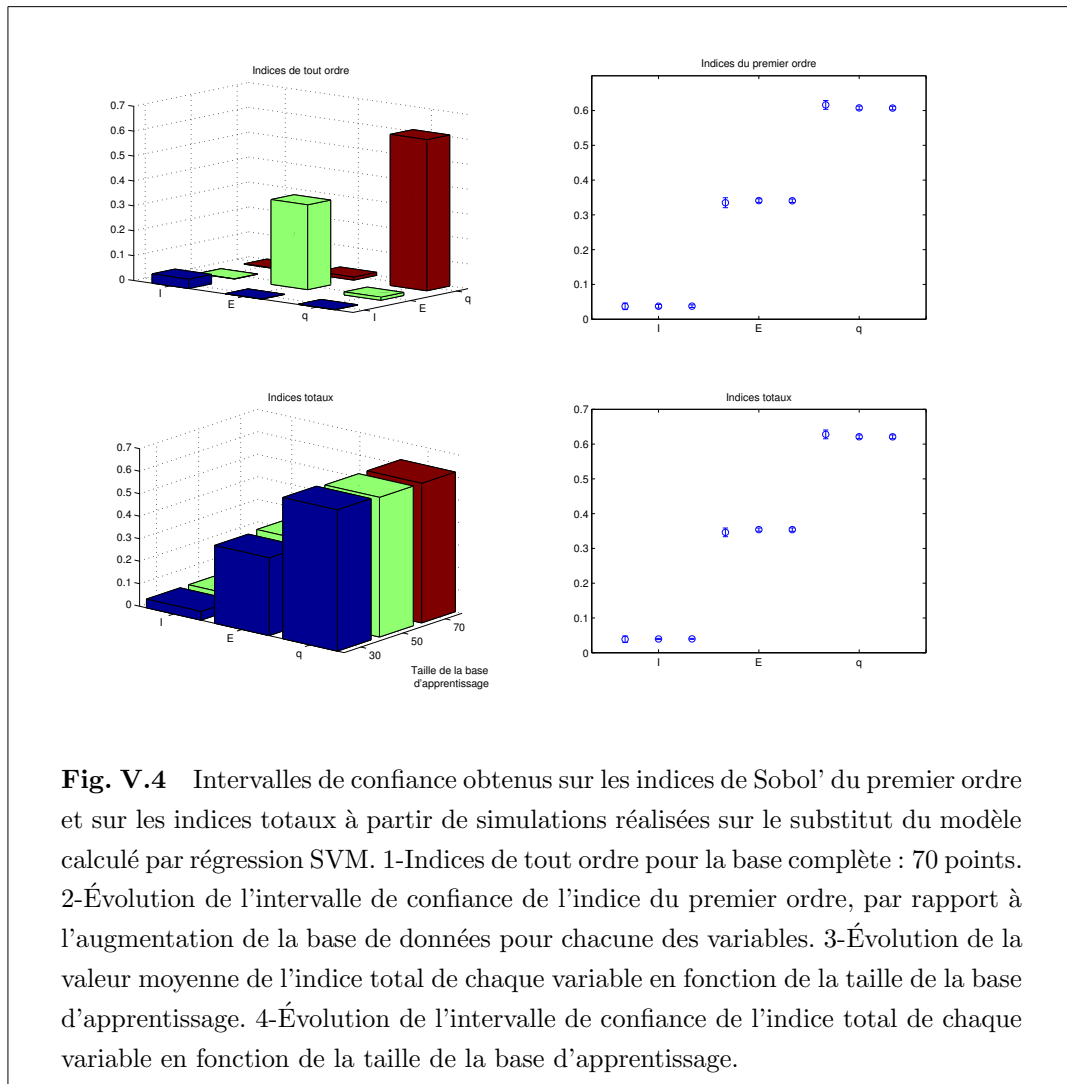
### Résultats de la régression

Les premiers résultats présentés, figure V.3, montrent que pour ce cas relativement simple, toutes les méthodes amènent des résultats équivalents, au moins sur les deux premiers moments. Pour les moments d'ordres supérieurs, la régression sur un polynôme d'ordre 2 est insuffisante.

Pour tous les moments, l'augmentation de la taille de la base de données permet de rendre l'intervalle de confiance de plus en plus serré autour de la valeur de référence.

La mesure des indices de Sobol', figure V.4, montre que, pour ce cas précis, 30 points générés par le carré Latin suffisent à l'obtention de résultats précis sur les indices de Sobol'. Ainsi 30 réalisations suffisent à la hiérarchisation des variables aléatoires d'entrée du modèle par régression SVM.

Ce premier exemple simple montre que la stratégie mise en œuvre est correcte, et permet de contrôler la régression par SVM sur différents exemples.



### V.4.5 Treillis

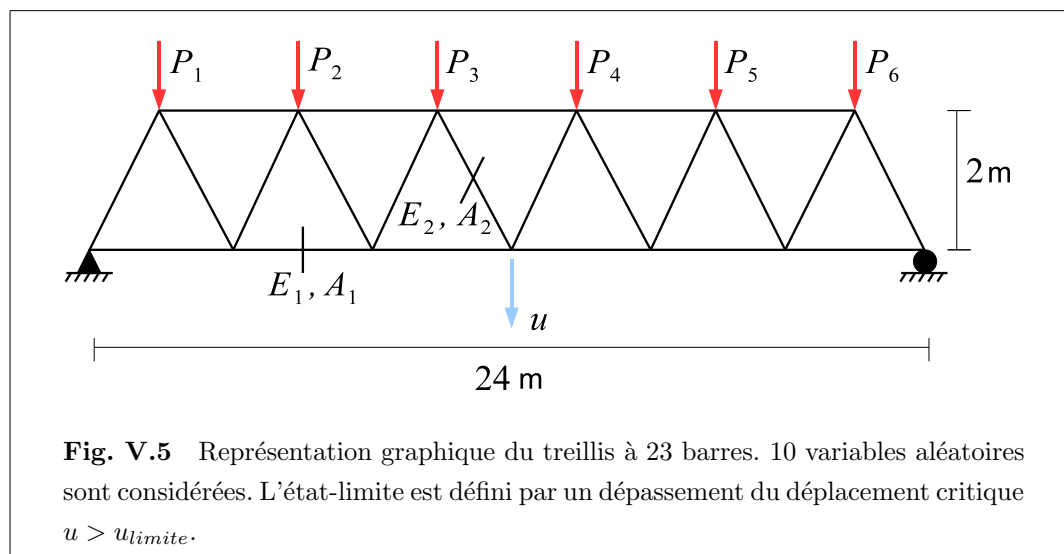
#### Description du problème

Cet exemple est issu du travail en collaboration avec EdF. G. Blatman [Bla06] propose ce treillis pour comparer différentes méthodes pour l'évaluation des sensibilités.

Considérons la structure présentée en figure V.5.

Le treillis à 23 barres est soumis aux 6 efforts  $P_1, \dots, P_6$ . On s'intéresse au déplacement maximal  $u$  d'un point de la barre inférieure. Le calcul de ce déplacement est réalisé par un code éléments finis.

10 variables aléatoires sont associées à cette structure :



- 6 définissent les efforts verticaux appliqués à chaque noeud supérieur ;
- 4 définissent les caractéristiques matériaux des barres horizontales et diagonales.

Les lois de ces paramètres sont définies dans le tableau V.2. L'état-limite est défini par un dépassement du déplacement critique  $u > u_{limite}$ , avec  $u_{limite} = 0,11$  cm.

Variables aléatoires	Loi	Moyenne	Écart-type
$E_1, E_2$ (MPa)	Lognormale	210	21
$A_1$ (m <sup>2</sup> )	Lognormale	$2 \cdot 10^{-3}$	$2 \cdot 10^{-4}$
$A_2$ (m <sup>2</sup> )	Lognormale	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$
$P_{1,...,6}$ (N)	Gumbel	$5 \cdot 10^4$	$7,5 \cdot 10^3$

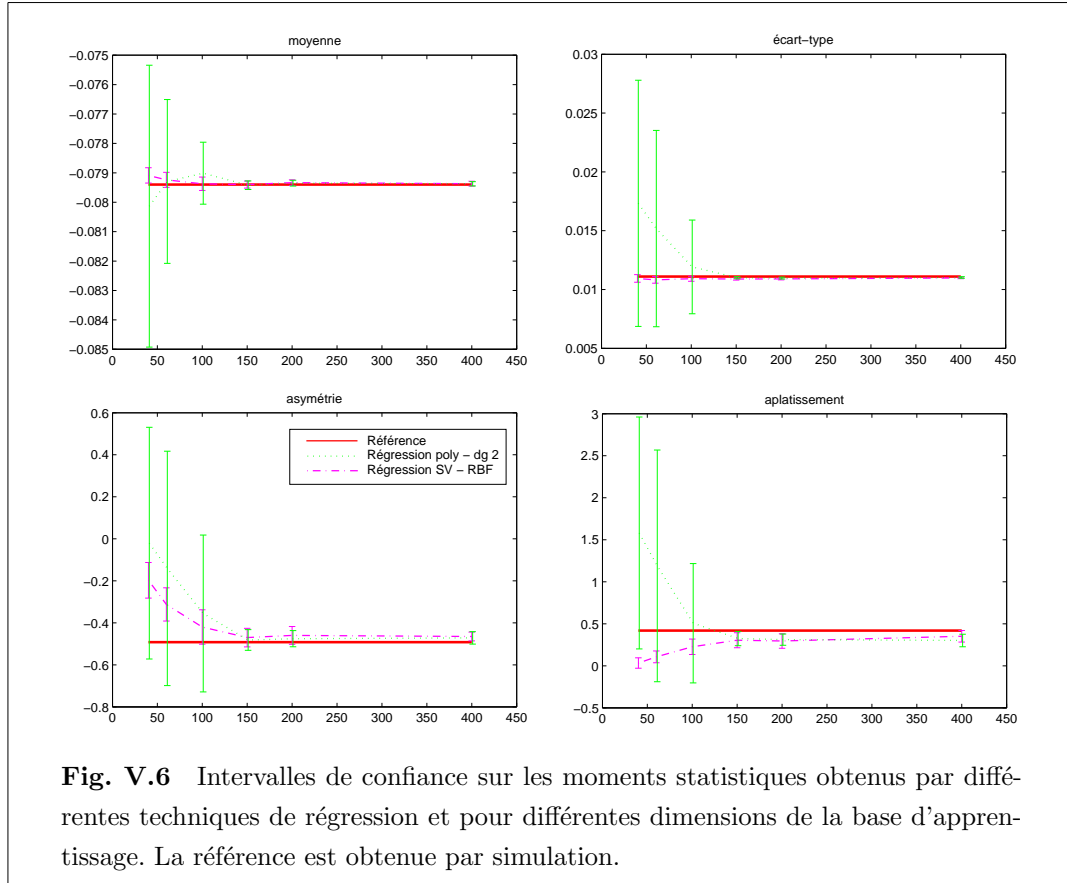
**Tab. V.2** Variables aléatoires pour le treillis.

### Résultats en régression

Étant donné la taille du problème, la régression par moindres carrés sur un polynôme de degré 3 est trop coûteuse en points d'apprentissage. Seule la régression SVM et la régression sur un polynôme de degré 2 sont comparées. La figure V.6 montre que la régression polynomiale est trop rigide pour permettre une bonne représentation des moments. Plus précisément, la valeur n'est pas excessivement éloignée, mais l'intervalle de confiance est beaucoup moins serré autour de la valeur de référence.

Pour les deux premiers moments, moyenne et écart-type, 40 points d'apprentissage

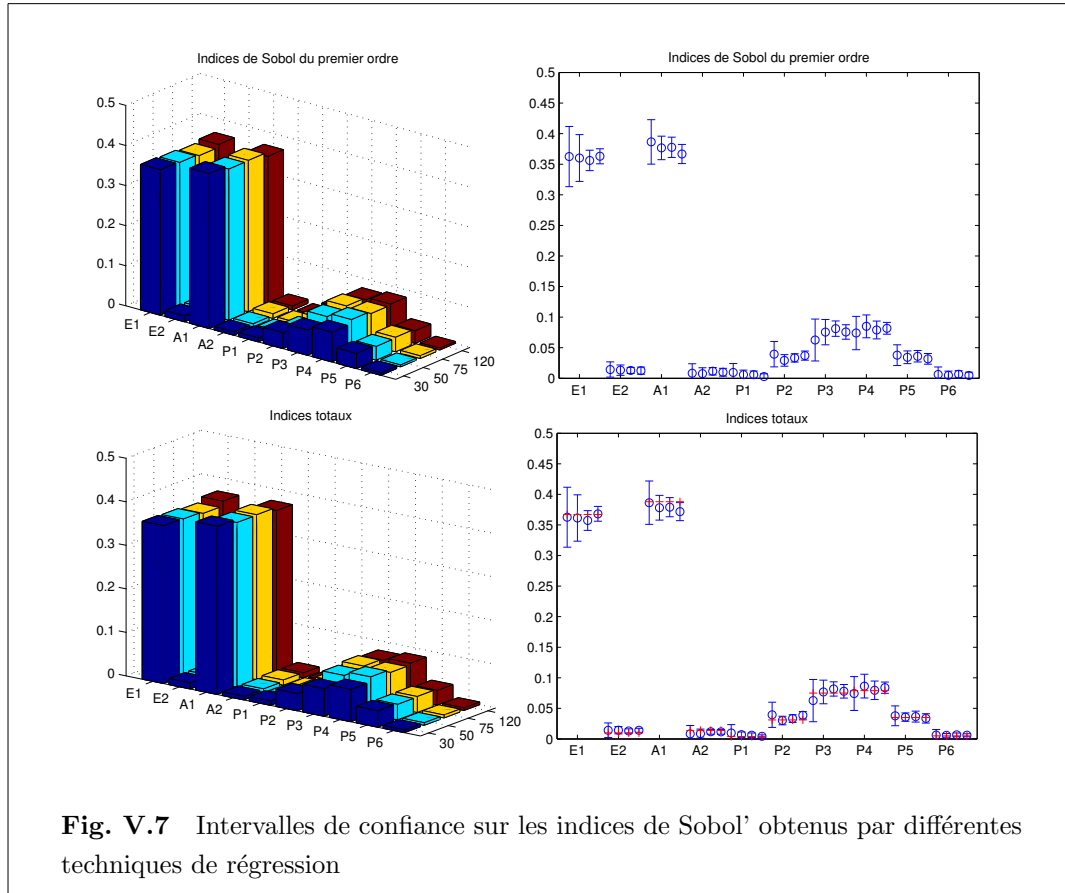
suffisent pour la régression SVM, alors que la régression polynomiale en nécessite au moins le double.



Concernant l'évaluation des indices de Sobol' par simulation sur la surface SVM, seul l'intervalle de confiance se resserre avec l'augmentation de la taille de la base de données. En effet, la valeur de l'ensemble des indices de Sobol' mesurés est suffisante à partir de 30 points d'apprentissage, malgré la dimension 10. Il n'y a aucun changement dans la hiérarchie des variables avec l'augmentation de la taille de la base.

#### V.4.6 Discussion

Globalement pour chacun des cas présentés, lorsque la base de données est assez large, les résultats obtenus par les différentes techniques sont satisfaisants et assez proches. Cependant les résultat SVM sont plus rapidement proches de la référence et avec un intervalle de confiance serré autour de la référence. En fait, lorsque la base d'apprentissage est de taille réduite, et lorsque la dimension du problème est moyenne



(autour de 10 variables), la régression polynomiale d'ordre élevé 3 est impossible, et le degré 2 est insuffisant pour représenter la fonction. Dans ce cas de figure, seule la régression par les SVM est possible et donne des résultats satisfaisants. Cette remarque est valable pour l'évaluation des premiers moments et pour les indices de Sobol'.

Plus la dimension augmente, plus la fonction est complexe, et plus la régression SVM est une réponse nécessaire en cas de base d'apprentissage réduite, puisque ce type de régression est moins sensible à la dimension. Il faut tout de même modérer cette conclusion par le fait que nous ne présentons ici que 2 exemples. D'autres essais sont proposés dans le chapitre VI.

Un problème subsiste cependant : la sélection des points d'apprentissage. Un début de réponse est proposé dans la suite.

## V.5 Méthode Sparse Grid

Jusqu'ici, le plan d'expériences utilisé pour définir la base d'apprentissage des points est réalisé par Monte-Carlo, carré latin ou encore par les points issus du plan généré par l'algorithme de classification. Cette dernière approche permet la réutilisation des informations obtenues par un calcul de fiabilité pour une exploitation en sensibilité.

La classification, appliquée à la construction de l'état-limite pour la fiabilité, a permis l'élaboration d'une méthode permettant de sélectionner à chaque étape de nouveaux points d'apprentissage presque optimaux. L'idéal serait de trouver une méthode comparable pour la régression.

C'est ce qui est proposé par l'approche en *Sparse Grid* associée à une optimisation de la construction du plan d'expériences ou de la grille, en fonction de l'erreur mesurée. Les sparse grids sont une technique permettant de représenter, d'intégrer ou d'interpoler des fonctions en grande dimension. À l'origine de cette technique, on trouve le mathématicien Smolyack. On trouve les généralités sur les grilles et leurs applications dans : [BG04b].

Cette section présente la technique dans ses grandes lignes et les résultats obtenus sur une première utilisation.

### V.5.1 L'interpolation par la méthode de la grille

L'interpolation par la méthode de la grille est un problème de couverture optimale : celui de sélectionner des points d'interpolation de manière à ce que la fonction soit approximée par une fonction d'interpolation optimale. La méthode d'interpolation de Smolyack initialement écrite en dimension 1 est étendue au cas de variables multiples. Nous présentons ici quelques types de grilles utiles pour l'interpolation.

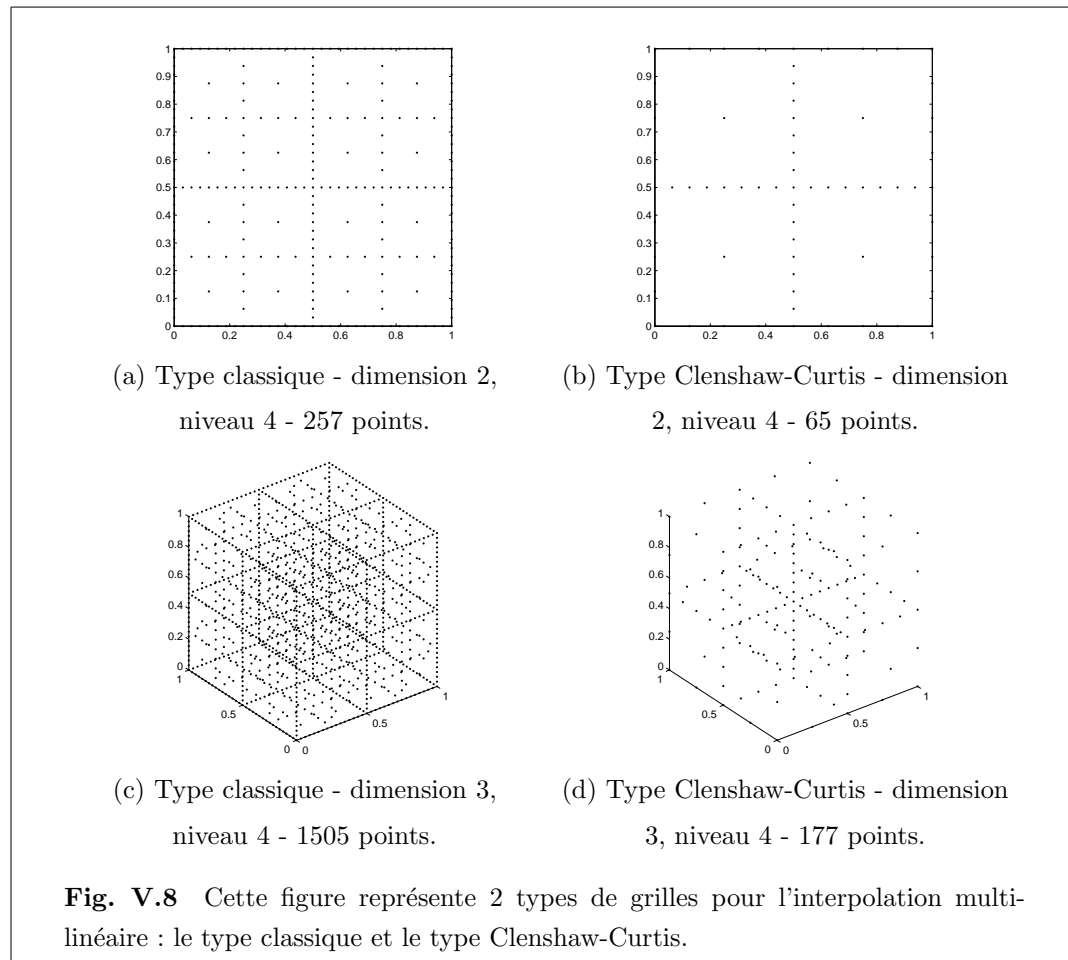
#### Smolyack

La règle de Smolyack permet de diminuer fortement le nombre de points nécessaires à l'interpolation en comparaison avec les méthodes de quadrature complète.

La figure V.8 présente deux types de grilles utilisées pour l'interpolation multilinéaire. Les grilles en dimension 2 et 3 sont représentées pour le niveau 4. Dans chacun des cas, la règle de quadrature montre que la grille d'ordre  $n - 1$  est incluse dans la grille d'ordre  $n$  ce qui permet d'améliorer l'approximation sans réécrire toutes

les expériences. Pour chaque type de grille, une règle déterminant le poids de chaque points est établie.

Il existe une formule permettant d'écrire l'erreur de l'interpolation *a priori*. Autrement dit, en se fixant un critère sur l'erreur, il est possible d'augmenter le niveau de la grille de manière incrémentale, jusqu'à la satisfaction du critère étudié.



### Interpolation polynomiale

L'interpolation multi-linéaire peut-être améliorée par l'utilisation de base de fonction polynomiale, par exemple les polynômes de Lagrange. Cette technique a été étudiée par Barbato *et al.* dans [BNR00].

Les grilles les plus adaptées à cette technique sont les grilles de Chebyshev-Gauss-Lovato. Le nombre de points de ces grilles est équivalent aux grilles de type Clenshaw-Curtis. La différence tient dans le fait que les points ne sont pas équidistants, cette ca-

ractéristique n'étant pas satisfaisante pour l'interpolation par des polynômes d'ordre supérieurs [Kli06b].

### V.5.2 Optimisation de la grille

L'intérêt de la méthode ne provient pas réellement de l'interpolation polynomiale. En effet, la grille complète, peu importe le type, devient trop importante lorsque la dimension augmente. En revanche, la sélection automatique optimale des points en fonction des dimensions importantes, voire des zones de l'espace mal approximées, devient extrêmement intéressante pour les problèmes en grande dimension.

Pour une grille standard, chaque direction est traitée de la même façon, soit avec le même nombre de points. Dans un cas où la dimension est importante, se pose alors la question de l'intérêt de traiter chaque variable de la même façon alors que leurs poids respectifs sont différents.

A partir des algorithmes proposés par T. Gernster et M. Griebel [GG03], puis en optimisant la méthode pour son implémentation, A. Klimke [KW05, Kli06a] élabore sous MATLAB une boîte à outils permettant l'optimisation des grilles pour l'interpolation polynomiale.

En quelques mots, A. Klimke définit un indice d'erreur indexé sur chaque direction du problème [Kli06b]. En priorité, la direction dans laquelle l'erreur est la plus importante est affinée prioritairement aux autres.

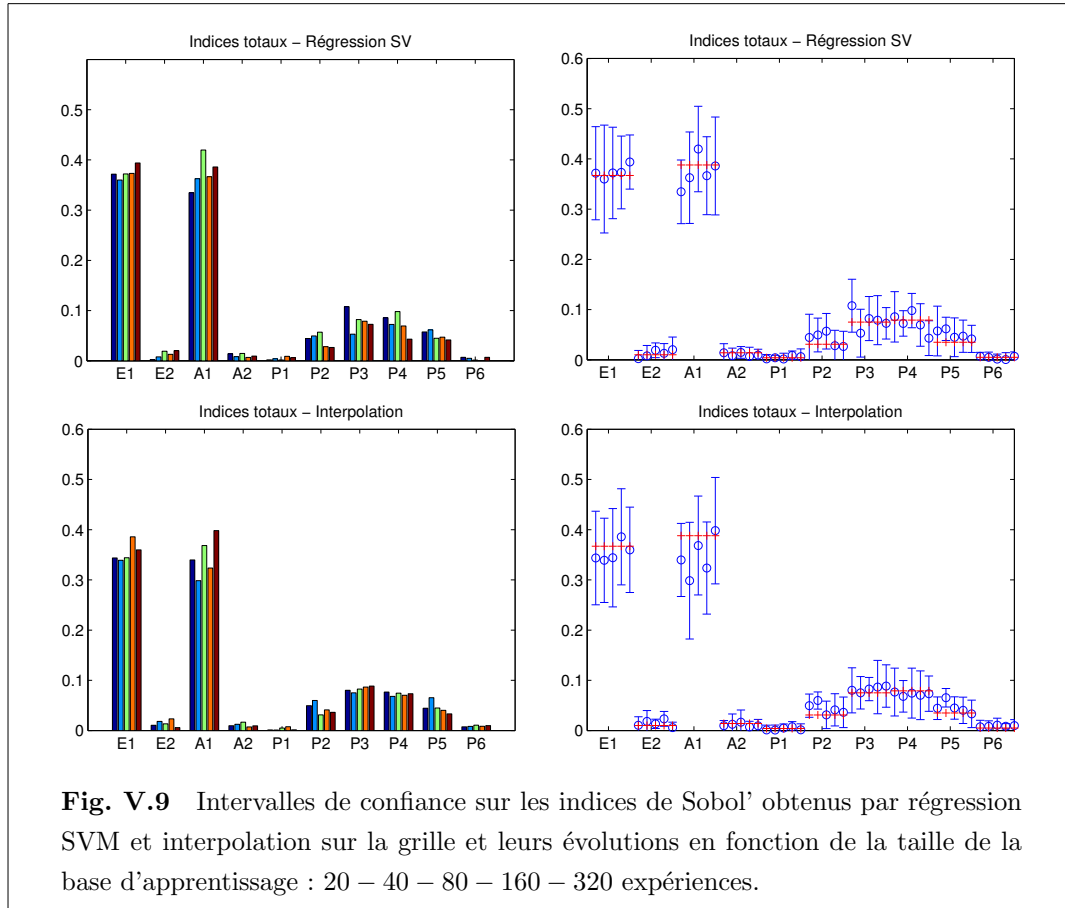
### V.5.3 Premiers résultats

La méthode sparse grid avec optimisation du choix des expériences en fonction des directions, par rapport à leur influence et à l'erreur obtenue localement, a été testée sur le cas du treillis.

Les indices de Sobol' totaux de chaque variable et les moments statistiques sont évalués par simulation sur la fonction obtenue par régression sur la base de donnée créée par l'algorithme d'optimisation de la grille. Les figures V.9 et V.10 présentent la comparaison des résultats obtenus par régression SVM et de ceux obtenus par interpolation polynomiale sur les mêmes expériences issues de la grille.

Les résultats obtenus par les 2 méthodes sont comparables. Il est intéressant de remarquer que la taille de la base utilisée ne permet pas la régression sur des polynômes de degré 2 ou 3, mais elle est cependant suffisante pour l'interpolation sur la grille. L'optimisation et le choix des expériences selon l'importance des variables,



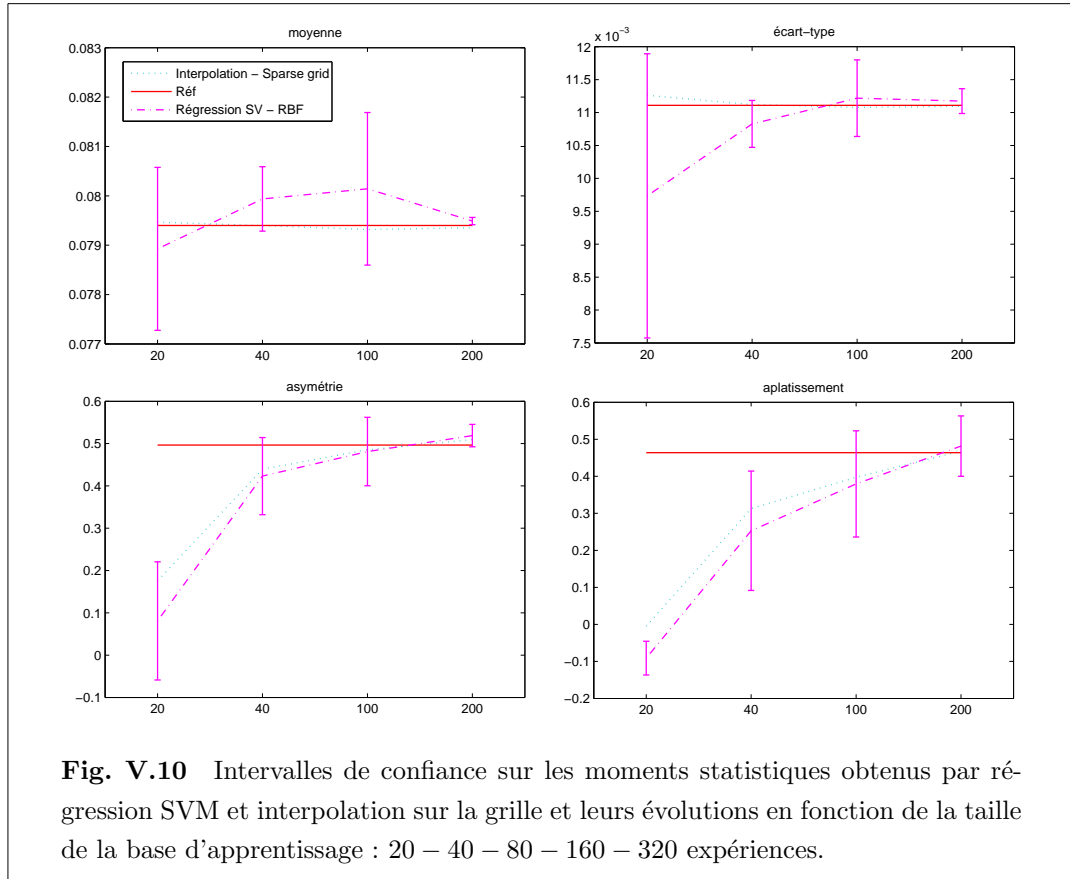


permettent de choisir un polynôme suffisamment souple et régulier à la fois. Le compromis complexité - généralisation est dans ce cas piloté par la grille, et il est ainsi adapté en fonction des directions.

#### V.5.4 Discussion

Les résultats obtenus par l'interpolation polynomiale sur la grille sont comparables à ceux obtenus par la régression SVM sur ces mêmes points. À degré de polynôme équivalent, et surtout à nombre de points équivalent pour l'apprentissage, l'interpolation par la grille est bien plus efficace que la régression polynomiale lorsque la dimension augmente (ici dimension 10). Ceci est sans doute à lier à l'optimisation de la grille sur la dimension.

La technique *sparse grid* permettrait de donner une règle de construction itérative de la base de données d'apprentissage dans le cadre de la régression. C'est un apport



très important pour la régression par rapport à ce qui est fait actuellement.

De la même manière que la marge SVM en classification permet l'optimisation du positionnement des points d'apprentissage pour la construction de la frontière entre défaillance et sûreté, les grilles adaptatives permettent la sélection optimale des points d'apprentissage pour la construction de la base de données pour la régression.

## V.6 Conclusion

Ce chapitre constitue un début de réponse au problème de la sensibilité, qui n'est pas un objectif premier de cette thèse. Sur les premiers exemples présentés, la flexibilité de l'approche SVM permet de résoudre le problème de la sensibilité de manière satisfaisante. Ceci est d'autant plus vrai en dimension importante alors que la régression polynomiale devient très coûteuse.

Le problème de la création de la base de données optimale pour la régression reste

entier. Les plans aléatoires sont satisfaisants mais ne permettent pas de plan élaboré de manière incrémentale. La réponse apportée par la technique de *sparse grid* est de ce point de vue très intéressante. Grâce à cette technique, les points sont choisis selon les directions les plus influentes ou selon les directions où l'erreur est la plus importante. De la même façon que la marge et la classification nous apportent une méthode optimale de sélection de nouveaux points d'apprentissage pour l'étude de fiabilité, la grille permet d'atteindre cet objectif pour un problème de régression dans le cadre d'une étude de sensibilité.

Les premiers tests effectués sont encourageants et méritent d'être approfondis.

## Chapitre VI

# Applications Mécaniques

Après avoir validé la méthode sur différents cas de la littérature, quelques applications mécaniques (plus ou moins complexes...) mettent en avant les avantages des développements sur les SVM.

## Sommaire

---

<b>VI.1 Introduction</b>	<b>163</b>
<b>VI.2 Exemples simples de la mécanique</b>	<b>164</b>
VI.2.1 Une sphère sous pression	164
VI.2.2 Un treillis	168
VI.2.3 Un oscillateur à 2 degrés de liberté primaire-secondaire	169
VI.2.4 Discussion	174
<b>VI.3 Plaque trouée</b>	<b>175</b>
VI.3.1 Description du modèle mécanique	175
VI.3.2 Résultats du calcul fiabiliste	176
VI.3.3 Plaque à 8 trous	177
<b>VI.4 Le projet PROSIR</b>	<b>182</b>
VI.4.1 Description du modèle mécanique	184
VI.4.2 Résultats fiabilistes	185
VI.4.3 Le défaut aléatoire	187
<b>VI.5 Conclusion</b>	<b>188</b>

---

## VI.1 Introduction

**I**L S'AGIT de montrer l'efficacité des méthodes développées sur plusieurs applications de la mécanique, applications présentant des difficultés de résolution plus ou moins conséquentes.

Le couplage avec un code par éléments finis est mis en place pour résoudre des modèles dont la complexité est située à différents niveaux :

- le nombre de variables aléatoires ;
- la forme de l'état-limite (linéarité / non-linéarité, continuité / discontinuité, point de défaillance unique / multiple).

Ces différents items perturbent la résolution d'un problème de fiabilité par les méthodes classiques. Nous montrons ici que les techniques d'apprentissage permettent de traiter avec succès les types de problématiques présentées.

Différentes applications sont ici étudiées :

1. quelques applications mécaniques simples : une cuve sphérique soumise à une pression interne aléatoire, un treillis à 23 barres et un oscillateur primaire-secondaire accordé à 2 degrés de liberté.
2. Une plaque métallique percée, dont les caractéristiques matériaux sont définies par un champ aléatoire. Cet exemple est tiré d'un rapport de Liu et Der Kiureghian, [LADK86]. Nous proposons un exemple dérivé constitué d'une plaque avec trous multiples.
3. Un exemple EdF traitant d'une cuve présentant un défaut de structure aléatoire. Il est issu du benchmark PROSIR. La complexité de l'état-limite est intéressante pour valider notre procédure de calcul, en effet de multiples points de conception entraînent des problèmes d'approximation FORM.

Résumons les difficultés pour chacune des applications :

- aucune en particulier pour la sphère et le treillis, si l'on exclut la difficulté induite par le couplage avec un code externe et son influence sur l'évaluation des gradients et donc la convergence de l'algorithme FORM.

- L'oscillateur primaire-secondaire accordé présente une probabilité de défaillance faible et une forte courbure de l'état-limite.
- Le problème de la plaque trouée est caractérisé par des points de défaillance multiple. Il en va de même pour le projet PROSIR.

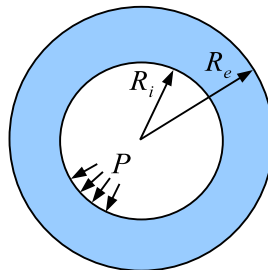
## VI.2 Exemples simples de la mécanique

Cette section est consacrée à l'étude d'exemples simples mais dont les équations proviennent de comportements de structures réelles. Le couplage avec un logiciel éléments finis est utilisé pour l'évaluation de la fonction d'état-limite. Ce couplage engendre une difficulté supplémentaire pour le calcul des gradients, et donc pour FORM.

### VI.2.1 Une sphère sous pression

#### Description du problème

Considérons la structure présentée sur la figure VI.1.



**Fig. VI.1** Représentation graphique de la sphère sous pression. L'état-limite considéré est un dépassement de contrainte de Von Mises admissible.

4 variables aléatoires (tableau VI.1) sont associées à la sphère sous pression :

- des variables géométriques : les rayons intérieur  $R_i$  et extérieur  $R_e$ ;
- une variable matériau : la limite d'élasticité  $f_e$ ;
- une variable de chargement : la pression interne  $P$  à laquelle est soumise la structure.

L'état-limite  $G$  est défini par le dépassement de la limite d'élasticité en peau interne,

qui est évalué de la manière suivante :

$$G(X) = f_e - \sigma(r = R_i) = f_e - \frac{1,5PR_e^3}{R_e^3 - R_i^3} \quad (\text{VI.1})$$

Variables aléatoires	Loi	Moyenne	Écart-type
$P$ (MPa)	Lognormale	130	8
$R_i$ (mm)	Lognormale	50	2,5
$R_e$ (mm)	Lognormale	100	5
$f_e$ (N)	Lognormale	300	20

**Tab. VI.1** [Sphère sous pression] Description des variables aléatoires.

### Les résultats fiabilistes

Les résultats obtenus par différentes méthodes sont présentés dans le tableau VI.2. La technique <sup>2</sup>SMART donne des résultats comparables aux méthodes de simulation, tout en étant plus performante en terme d'appels à l'état-limite. Sur cet exemple, l'approximation FORM est satisfaisante, et étant donné le faible nombre de variables, cette approximation au premier ordre reste la plus efficace.

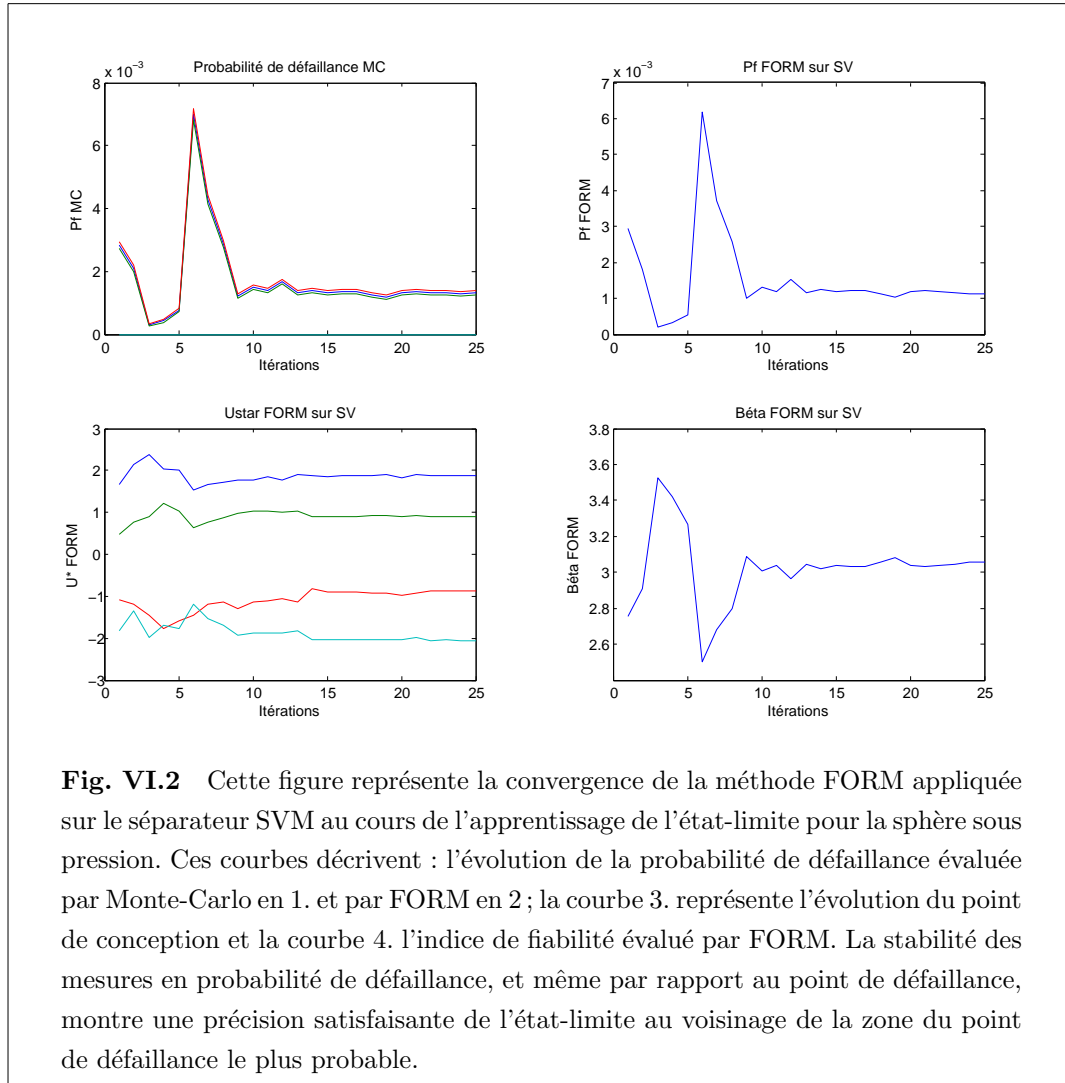
La figure VI.2 montre le suivi de la probabilité de défaillance évaluée par Monte-Carlo sur le séparateur SVM, en fonction des itérations de l'apprentissage. L'évaluation FORM sur le séparateur SVM est également présentée. Une rapide stabilisation des différents critères mesurés est obtenue.

Les résultats montrent que l'approximation de l'état-limite par une surface SVM est suffisamment précise globalement, pour l'évaluation de la probabilité de défaillance par simulation ; mais également localement, pour la convergence de l'évaluation du point de défaillance par FORM.

	MC	FORM	SORM	SS(AB)	SMART
$N$ appels	$10^6$	20	30	30000	120
$P_f$ ( $10^{-3}$ )	1,28	1,13	1,32	1,26	1,30
$\beta$	3,015	3,055	3,007	3,021	3,012
cv	0,28	—	—	0,07	0,26

**Tab. VI.2** [Sphère sous pression] Résultats du calcul fiabiliste.

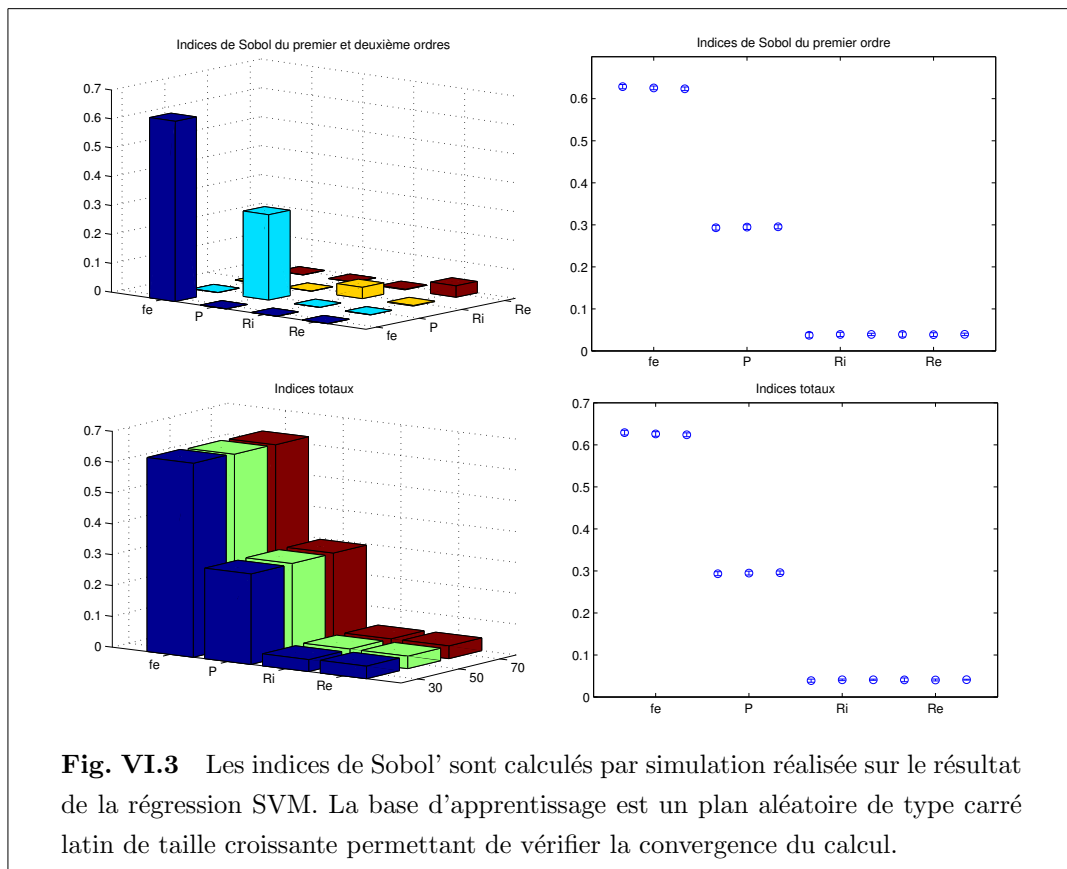




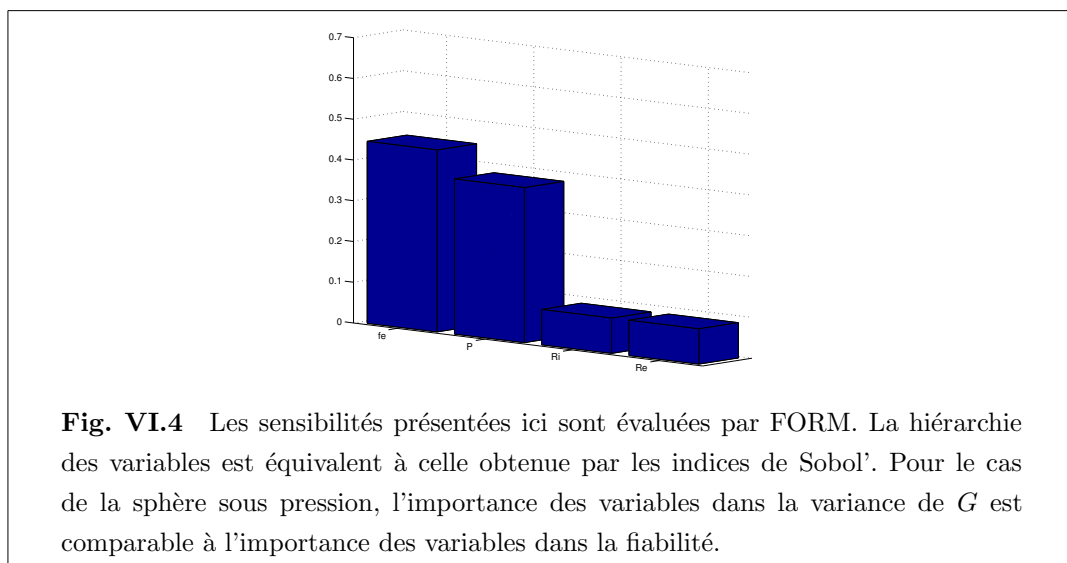
### Les résultats en sensibilité

Utilisons cet exemple pour vérifier les résultats en sensibilité (indices de Sobol') calculés par simulation sur le modèle SVM obtenu par régression sur une base construite par un plan de type carré latin.

La figure VI.3 présente les résultats obtenus et leur évolution en fonction de la dimension de la base de données utilisée. Sur ce cas très simple, 30 points suffisent à déterminer les indices de Sobol' de manière satisfaisante. La figure VI.4 présente les sensibilités FORM. La hiérarchie des variables est respectée entre les sensibilités mesurées par une analyse de la variance et la sensibilité à la fiabilité.



**Fig. VI.3** Les indices de Sobol' sont calculés par simulation réalisée sur le résultat de la régression SVM. La base d'apprentissage est un plan aléatoire de type carré latin de taille croissante permettant de vérifier la convergence du calcul.



**Fig. VI.4** Les sensibilités présentées ici sont évaluées par FORM. La hiérarchie des variables est équivalente à celle obtenue par les indices de Sobol'. Pour le cas de la sphère sous pression, l'importance des variables dans la variance de  $G$  est comparable à l'importance des variables dans la fiabilité.

### VI.2.2 Un treillis

Le treillis présenté section V.4.5 est repris pour un calcul de fiabilité. Rappelons que l'exemple est défini par 10 variables aléatoires, que l'état-limite est défini par un déplacement dépassant un seuil donnée, et que ce déplacement est évalué par éléments finis.

#### Les résultats fiabilistes

	MC	FORM	SORM	SS(AB)	<sup>2</sup> SMART
$N$ appels	$10^6$	66	72	30000	140
$P_f$ ( $10^{-3}$ )	8,75	5,00	9,08	9,44	8,75
$\beta$	2,38	2,57	2,36	2,35	2,38
cv	0,01	—	—	0,08	0,07

**Tab. VI.3** Résultats du calcul fiabiliste pour le cas du treillis.

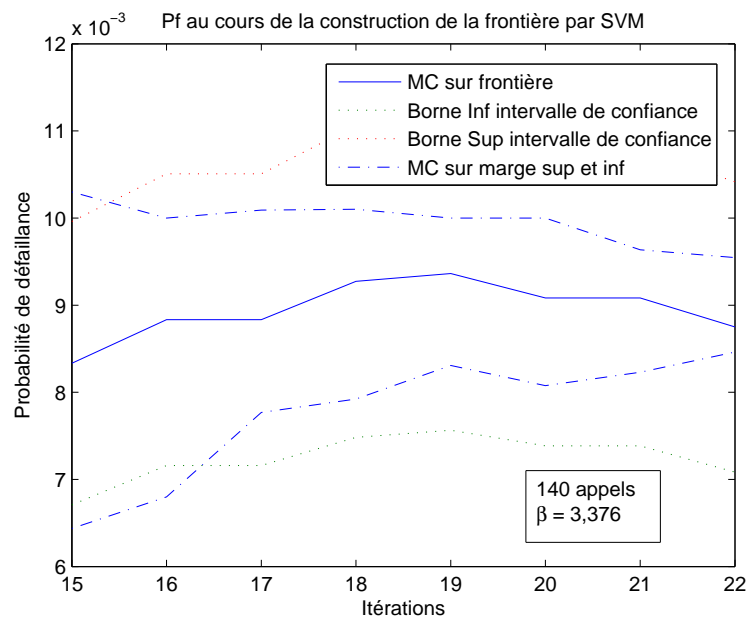
Les résultats obtenus par différentes méthodes sont comparés dans le tableau VI.3. Les approches par simulation de type *subsets* et le résultat par apprentissage sont très proches. Le décalage entre la valeur de la probabilité de défaillance FORM et la valeur obtenue par simulation est due à la courbure de l'état-limite au niveau du point de défaillance le plus probable comme l'atteste la valeur de l'estimation SORM.

La figure VI.5 présente 2 types d'intervalles autour de la valeur de la probabilité de défaillance :

- un intervalle de confiance à 90% compte tenu du nombre de simulations de Monte-Carlo faites sur le séparateur SVM pour l'évaluation de la probabilité de défaillance.
- un intervalle déterminé par simulation de Monte-Carlo sur les marges supérieure et inférieure du séparateur SVM.

Ces intervalles (figure VI.5) montrent que l'incertitude sur l'évaluation de la probabilité de défaillance par Monte-Carlo (méthode de référence) est supérieure à l'incertitude sur le séparateur.

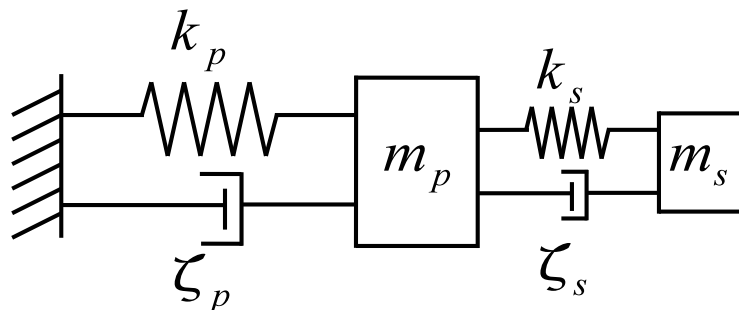
Des simulations réalisées sur les frontières inférieure et supérieure de l'état-limite encadrent le résultat en probabilité de défaillance : celui-ci est inclus dans l'intervalle de confiance défini par le niveau de probabilité estimé et le nombre de simulations, voir figure VI.5.



**Fig. VI.5** Cette figure présente la convergence de l'évaluation de la probabilité de défaillance sur la formulation analytique de l'état-limite. La simulation sur la marge donne un intervalle inférieur à celui que définit l'intervalle de confiance sur la simulation de Monte-Carlo.

### VI.2.3 Un oscillateur à 2 degrés de liberté primaire-secondaire

#### Description du problème



**Fig. VI.6** Schéma de l'oscillateur à 2 degrés de liberté primaire-secondaire.

Issu de la littérature, [DSDK90], cet autre exemple permet de mesurer la stabilité de la procédure proposée, face à un système dont l'état-limite est fortement courbé et donc dont la convergence FORM pose problème.

8 variables aléatoires sont associées à ce problème de dynamique. Leurs caractéristiques statistiques sont précisées dans le tableau VI.4.

La formule de l'état-limite est la suivante :

$$G(X) = F_s - 3k_s \sqrt{\pi \frac{S_0}{4\zeta_s \omega_s^3} \frac{\zeta_a \zeta_s}{\zeta_p \zeta_s (4\zeta_a^2 + \theta^2) + \gamma \zeta_a^2} \frac{(\zeta_p \omega_p^3 + \zeta_s \omega_s^3) \omega_p}{4\zeta_a \omega_a^4}} \quad (\text{VI.2})$$

où :

$$\begin{aligned} \omega_p &= \sqrt{\frac{k_p}{m_p}} \quad \text{et} \quad \omega_s = \sqrt{\frac{k_s}{m_s}} \\ \gamma &= \frac{m_s}{m_p} \\ \omega_a &= \frac{1}{2}(\omega_p + \omega_s) \\ \zeta_a &= \frac{1}{2}(\zeta_p + \zeta_s) \\ \theta &= \frac{1}{\omega_a}(\omega_p - \omega_s) \end{aligned} \quad (\text{VI.3})$$

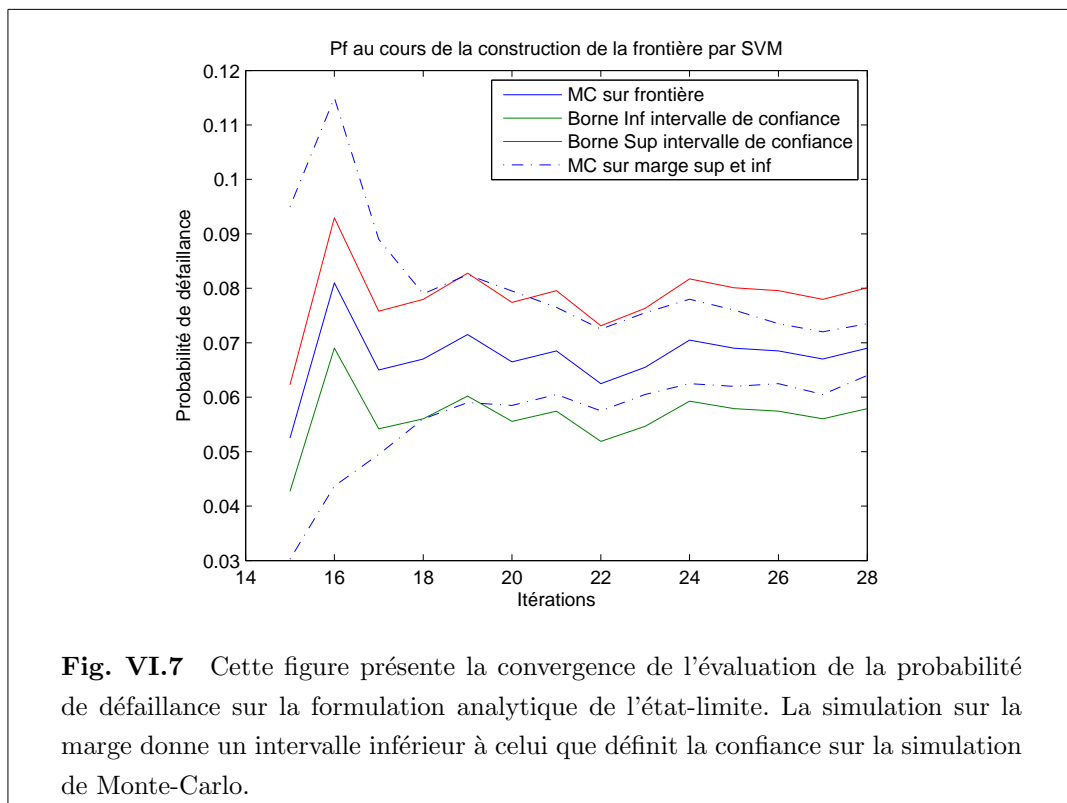
Variables aléatoires	Loi	Moyenne	CV
$m_p$	Lognormale	1,5	10%
$m_s$	Lognormale	0,01	10%
$k_p$	Lognormale	1	20%
$k_s$	Lognormale	0,01	20%
$\zeta_p$	Lognormale	0,05	40%
$\zeta_s$	Lognormale	0,02	50%
$F_s$	Lognormale	15	10%
$S_0$	Lognormale	100	10%

**Tab. VI.4** Variables aléatoires pour l'oscillateur non-linéaire.

### Les résultats fiabilistes

	MC	FORM	SS(AB)	<sup>2</sup> SMART
$N$ appels	$10^6$	2600	30000	700
$P_f$ ( $10^{-3}$ )	4,81	21,8	9,44	5,30
$\beta$	2,589	2,016	2,3480	2,555
cv	0,02	—	0,08	0,02

**Tab. VI.5** Résultats du calcul fiabiliste pour l'oscillateur non-linéaire.



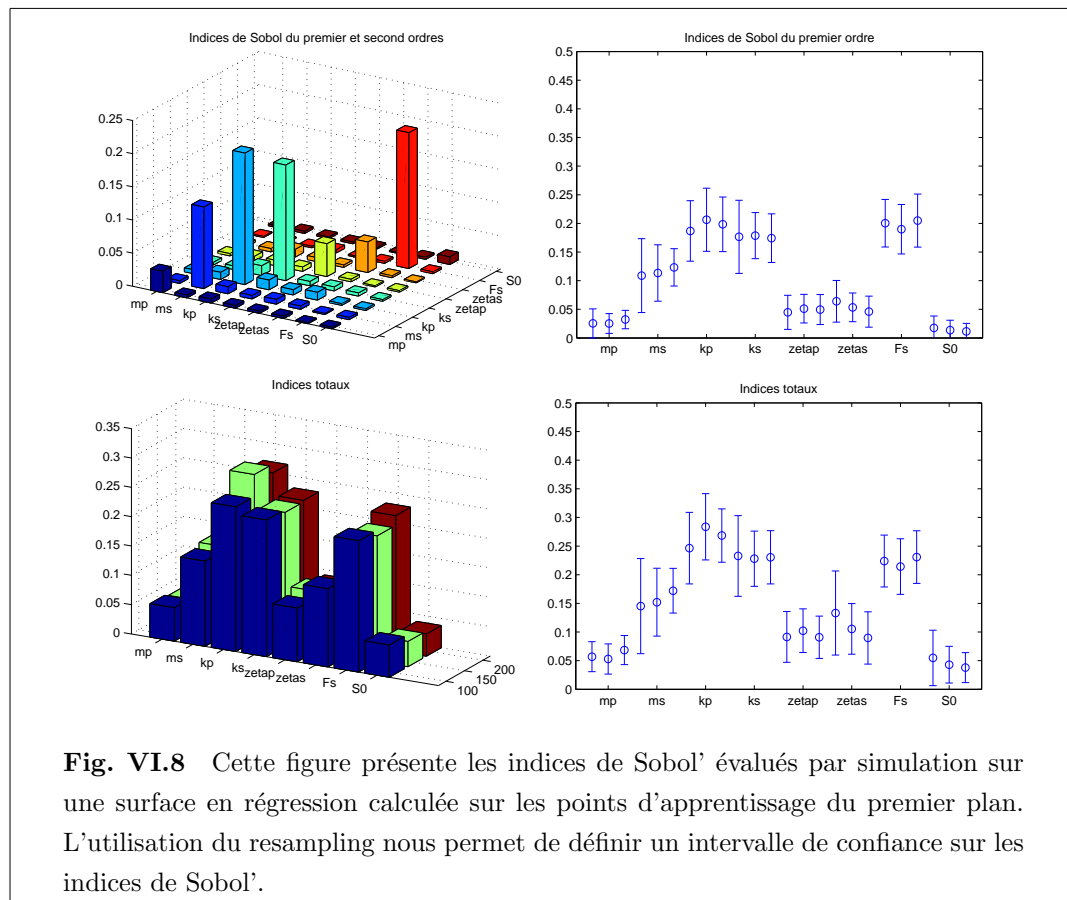
La valeur de la probabilité de défaillance obtenue par <sup>2</sup>SMART est très satisfaisante comparée aux valeurs mesurées par *subsets*. Le nombre de calculs diminue de manière drastique par rapport aux méthodes de simulations. L'algorithme FORM converge difficilement sur ce problème pour des probabilités de défaillance très faibles.

### Évaluation des indices de Sobol'

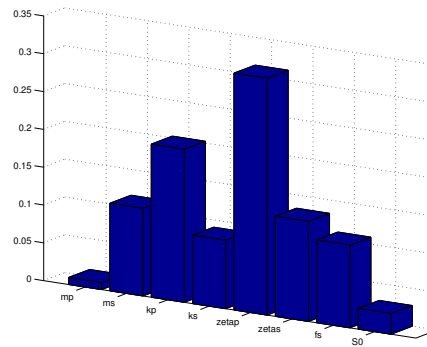
Les points d'apprentissage du premier plan d'expériences aléatoires permettent le calcul des indices de Sobol' sur la surface en régression (environ 100 points d'apprentissage). Les résultats sont présentés en figure VI.8. On remarque peu de différences entre les indices du premier ordre et les indices totaux. Les indices du deuxième ordre sont négligeables par rapport aux indices du premier ordre.

L'intervalle de confiance obtenu par resampling sur la base d'apprentissage montre que les valeurs obtenues sont acceptables. La hiérarchisation des variables par leur influence sur la variabilité de l'état-limite est possible par régression sur la base d'apprentissage de l'algorithme utilisé pour l'étude de fiabilité.

La figure VI.9 présente les indices de sensibilités obtenus par FORM. Des différences apparaissent entre les sensibilités généralisées de Sobol' et les sensibilités à la défaillance FORM, qui sont des indices de sensibilité locale.



**Fig. VI.8** Cette figure présente les indices de Sobol' évalués par simulation sur une surface en régression calculée sur les points d'apprentissage du premier plan. L'utilisation du resampling nous permet de définir un intervalle de confiance sur les indices de Sobol'.



**Fig. VI.9** Les sensibilités présentées ici sont évaluées par FORM.

### Influence de la valeur du seuil

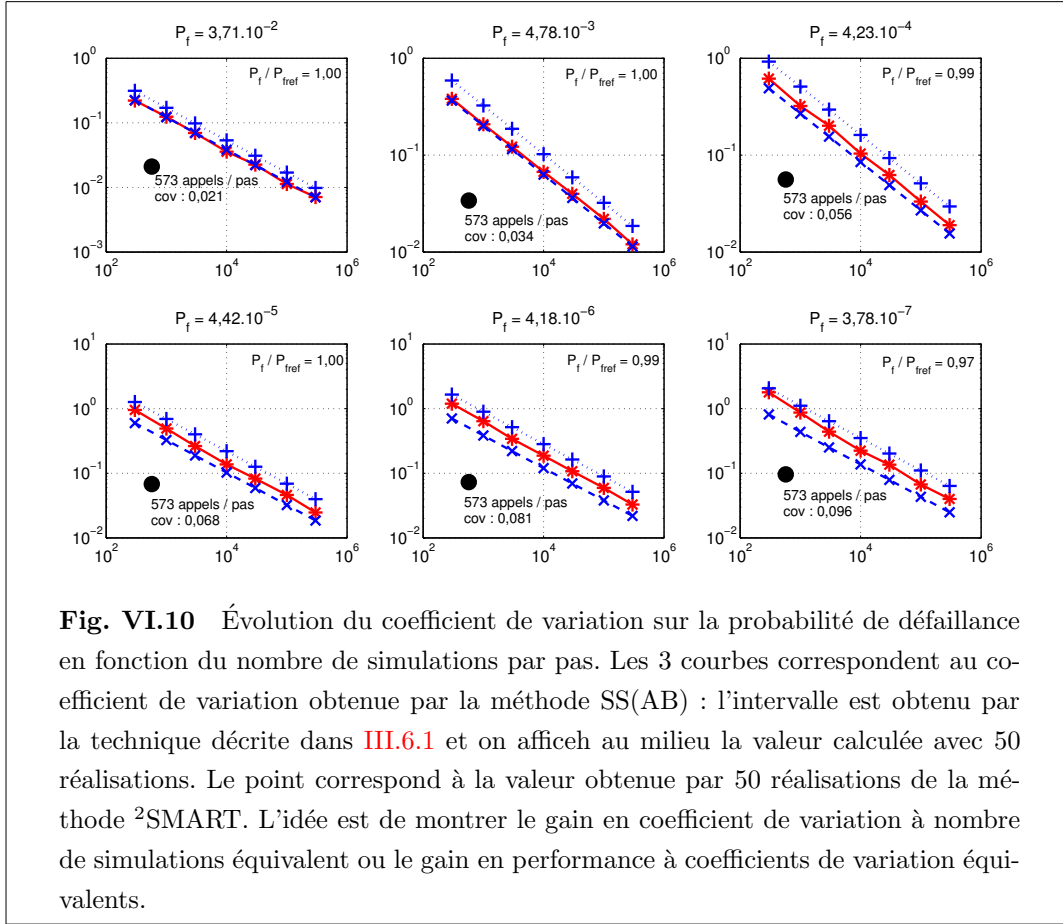
FORM	MC (100 calculs)	SS(AB) (500 calculs)	<sup>2</sup> SMART (50 calculs)
$P_f$ ( $N$ appels)	$P_f$ ( $N$ appels/cv %)	$P_f$ ( $N$ appels/cv %)	$P_f$ ( $N$ appels/cv %)
$9.70 \cdot 10^{-2}$ (234)	$3.71 \cdot 10^{-2}$ (27000 / 3.2)	$3.71 \cdot 10^{-2}$ ( $2 \times 3.10^5$ / 3.2)	$3.69 \cdot 10^{-2}$ ( $2 \times 300$ / 2.1)
$2.19 \cdot 10^{-2}$ (1179)	$4.79 \cdot 10^{-3}$ (200000 / 2.9)	$4.78 \cdot 10^{-3}$ ( $3 \times 3.10^5$ / 1.2)	$4.78 \cdot 10^{-3}$ ( $3 \times 300$ / 3.4)
$2.72 \cdot 10^{-3}$ (3099)	$4.22 \cdot 10^{-4}$ (2300000 / 3.1)	$4.23 \cdot 10^{-4}$ ( $4 \times 3.10^5$ / 1.9)	$4.18 \cdot 10^{-4}$ ( $4 \times 300$ / 5.6)
div.	-	$4.42 \cdot 10^{-5}$ ( $5 \times 3.10^5$ / 2.5)	$4.42 \cdot 10^{-5}$ ( $5 \times 300$ / 6.8)
div.	-	$3.78 \cdot 10^{-7}$ ( $7 \times 3.10^5$ / 4.0)	$3.66 \cdot 10^{-7}$ ( $7 \times 300$ / 9.6)

**Tab. VI.6** Résultats du calcul fiabiliste pour l'oscillateur non-linéaire avec variation du seuil de résistance.

Une variation du seuil  $F_s$  a permis de tester notre méthode sur des probabilités de défaillance plus faibles. La stabilité de la solution a également été testée par une série de 50 lancements de la méthode <sup>2</sup>SMART. Cette étude a permis de montrer que sur ce cas non trivial, aucun biais n'a été détecté (cf. tableau VI.6).

Le coût de calcul par rapport au coefficient de variation obtenu est comparé pour





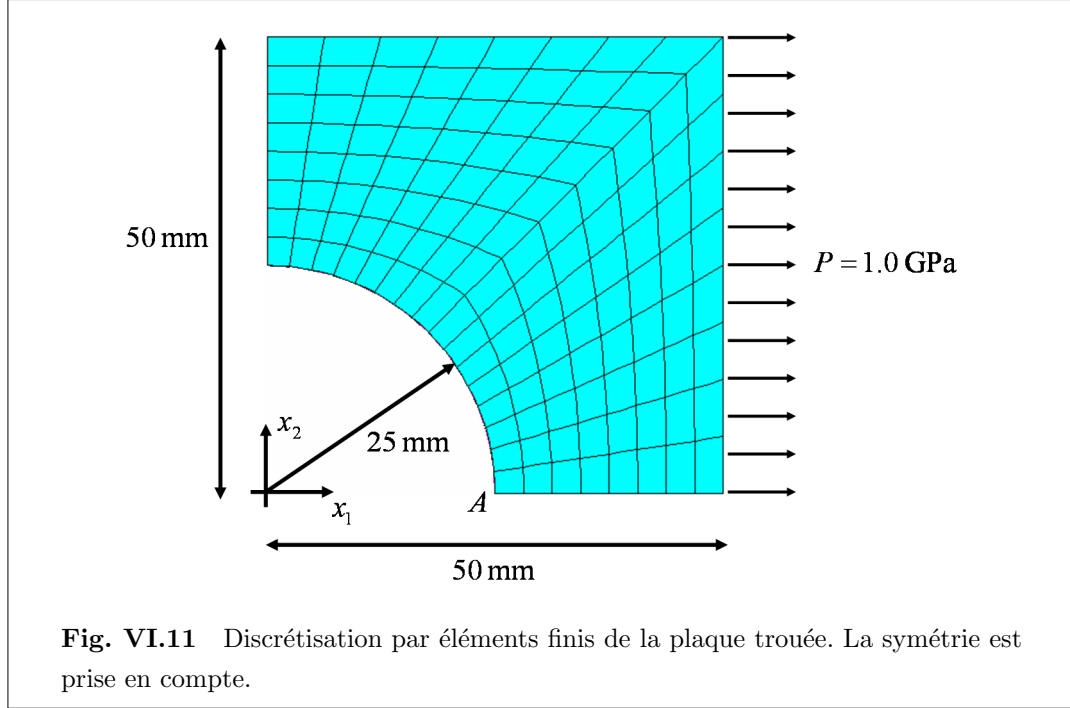
la méthode <sup>2</sup>SMART et la méthode des *subsets* sur le cas de l'oscillateur et pour différents seuils de probabilité, (cf. figure VI.10). Cette figure permet de montrer le gain en performance : soit par rapport au coefficient de variation à nombre de calculs donné, soit par rapport au nombre de calculs à coefficient de variation donné.

#### VI.2.4 Discussion

Ces premiers exemples de la mécanique permettent de valider l'approche du couplage des simulations avec l'apprentissage statistique, et en particulier la classification, pour la construction des frontières. La régression par les SVM est utile pour le calcul des indices de Sobol' et la hiérarchisation des variables. Dans la majorité des cas, la régression sur la base construite pour la fiabilité suffit au calcul des sensibilités.

## VI.3 Plaque trouée

### VI.3.1 Description du modèle mécanique



Une charge uniforme  $P$  est appliquée sur l'extrémité de la plaque représentée sur la figure VI.11. Le matériau est considéré dans ce cas élastique linéaire avec un coefficient de Poisson d'une valeur de  $\nu = 0,3$ . Le module de Young est assimilé à un champ aléatoire lognormal de moyenne  $M = 200$  GPa et d'écart-type  $S = 50$  GPa. (L'exemple cité utilise un champ normal, mais le champ lognormal permet d'éviter le cas de réalisations négatives). La fonction de corrélation est définie comme suit :

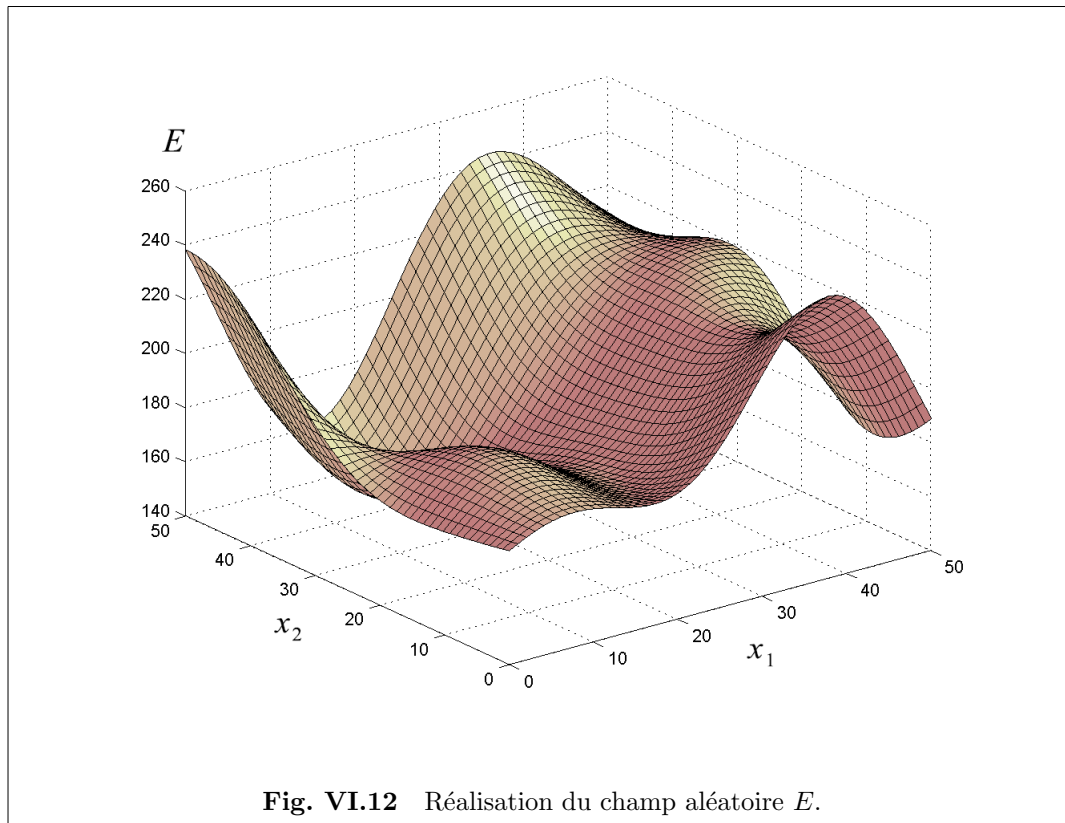
$$\rho = \exp\left(-\left(\frac{\Delta}{b}\right)^2\right) \quad (\text{VI.4})$$

où  $\Delta$  est la distance entre 2 points de la plaque et  $b = 2$  cm est la longueur de corrélation. Le champ aléatoire est discrétisé par le maillage par éléments finis présenté sur la figure VI.11 et la méthode *EOLE*, [SDK00]. Pour cette méthode, il faut qu'il y ait adéquation entre le maillage EF et la longueur de corrélation (un facteur 5 est pris ici). 20 variables aléatoires sont utilisées pour la représentation du champ avec une précision suffisante. La fonction d'état-limite est définie par la formule suivante :

$$G(X) = \theta - u(X) \quad (\text{VI.5})$$

où le vecteur  $X$  est composé des variables aléatoires définissant le champ  $E$ ,  $u(X)$  est le déplacement dans la direction  $x_1$  au point  $A$  et  $\theta = 0,19$  cm est le déplacement limite.

La figure VI.12 représente une réalisation du champ aléatoire  $E$ .



### VI.3.2 Résultats du calcul fiabiliste

Les résultats présentés dans le tableau VI.7 sont obtenus par différentes méthodes de fiabilité.

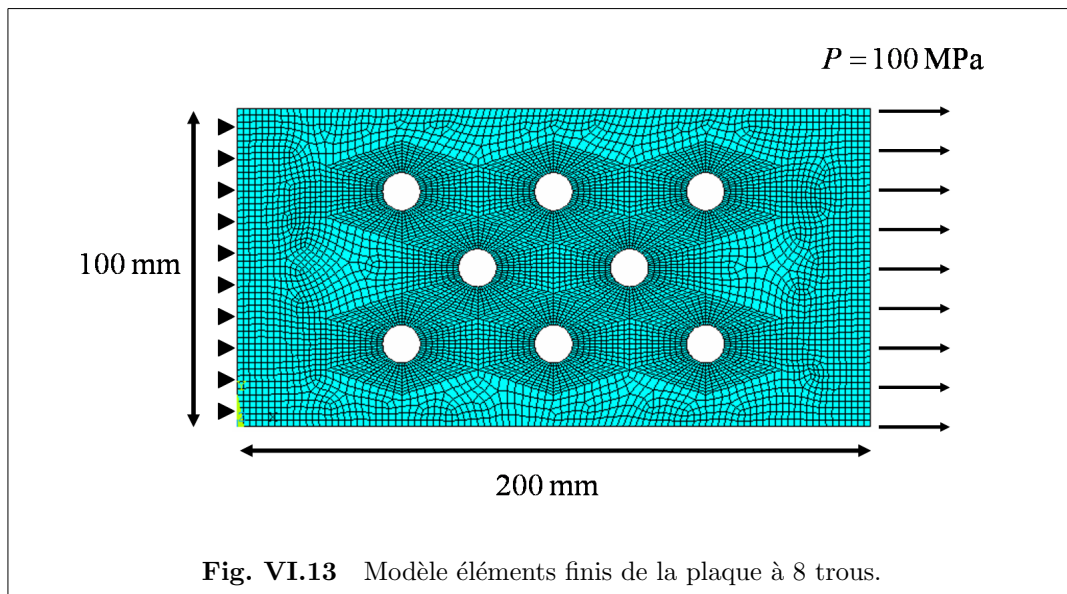
Pour ce modèle de taille relativement importante, la méthode <sup>2</sup>SMART donne des résultats comparables aux résultats de simulation pour un nombre d'appels à l'état-limite de l'ordre de 500.

	MC	FORM	SS(AB)	SS(AB)	<sup>2</sup> SMART
$N$ appels	$10^7$	66	5000	50000	500
$P_f$ ( $10^{-5}$ )	1,60	1,01	2,51	1,23	1,60
$\beta$	4,16	4,26	4,05	4,22	4,16
cv	0,25	-	0,34	0,12	0,03

**Tab. VI.7** Résultats de fiabilité pour le modèle de la plaque trouée en traction.  
Le module de Young  $E$  est ici défini par un champ lognormal.

### VI.3.3 Plaque à 8 trous

Le concept précédent de la plaque trouée est conservé, mais le modèle mécanique considéré est plus complexe. L'aspect symétrie du champ, physiquement non représentatif mais simplificateur du calcul, est maintenant abandonné. La discrétisation par éléments finis du modèle retenu est présentée figure VI.13.



Comme précédemment, le module de Young est défini par un champ aléatoire lognormal. L'état-limite est défini par une limite de contrainte équivalente sur toute la surface de la plaque. Cela permet d'obtenir de nombreux modes de défaillance et ainsi de vérifier le comportement de notre méthode face aux aspects système. Le problème contient ici plusieurs points de défaillance, de manière évidente. L'état-

limite est défini par la fonction suivante :

$$G(X) = \sigma_{limite} - \max(\sigma_{VonMises})$$

(VI.6)

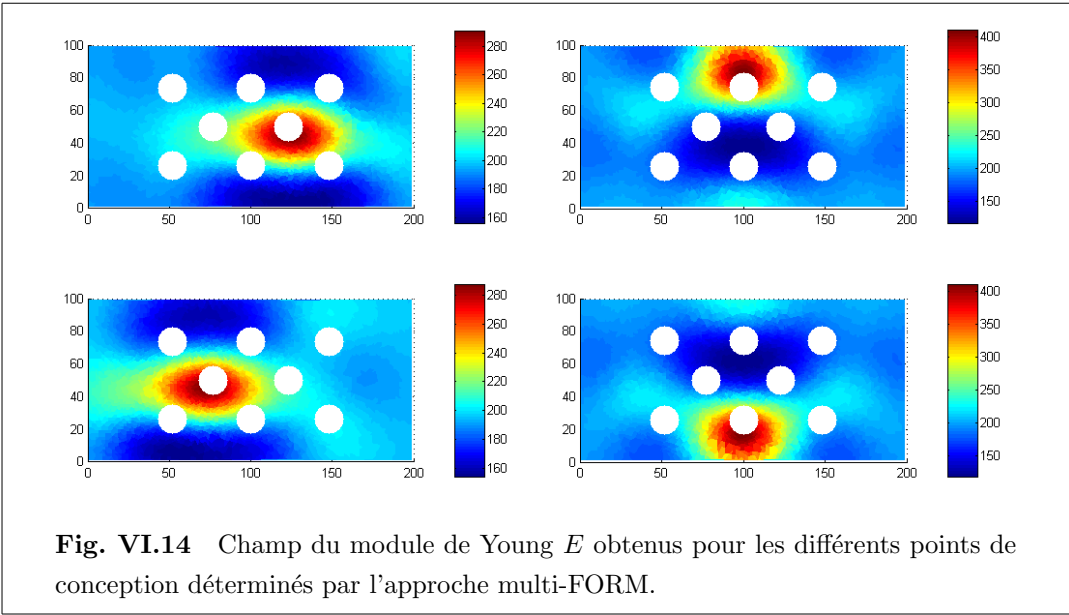
où,  $\sigma_{limite} = 450 \text{ MPa}$  est la contrainte limite et  $\max(\sigma_{VonMises})$  est la valeur maximale de la contrainte de Von Mises, aux nœuds, sur l'ensemble de la plaque.

Résultats du calcul fiabiliste

La probabilité de défaillance est évaluée par différentes méthodes. Les résultats obtenus sont collectés dans le tableau VI.8. Les résultats de l'analyse FORM, ne prenant en compte qu'un seul mode de défaillance, c'est-à-dire un seul point de défaillance, sont éloignés du résultat de référence, calculé par *subsets*. Une approche multi-FORM a été testée. Quatre modes de défaillance sont observés grâce à cette technique. Ils sont représentés sur la figure VI.14.

	FORM	SS(AB)	SS(AB)	<sup>2</sup> SMART
$N$ appels	200	2000	20000	600
$P_f$ ( $10^{-2}$ )	0,37	1,44	1,43	1,38
$\beta$	2,67	2,18	2,19	2,21

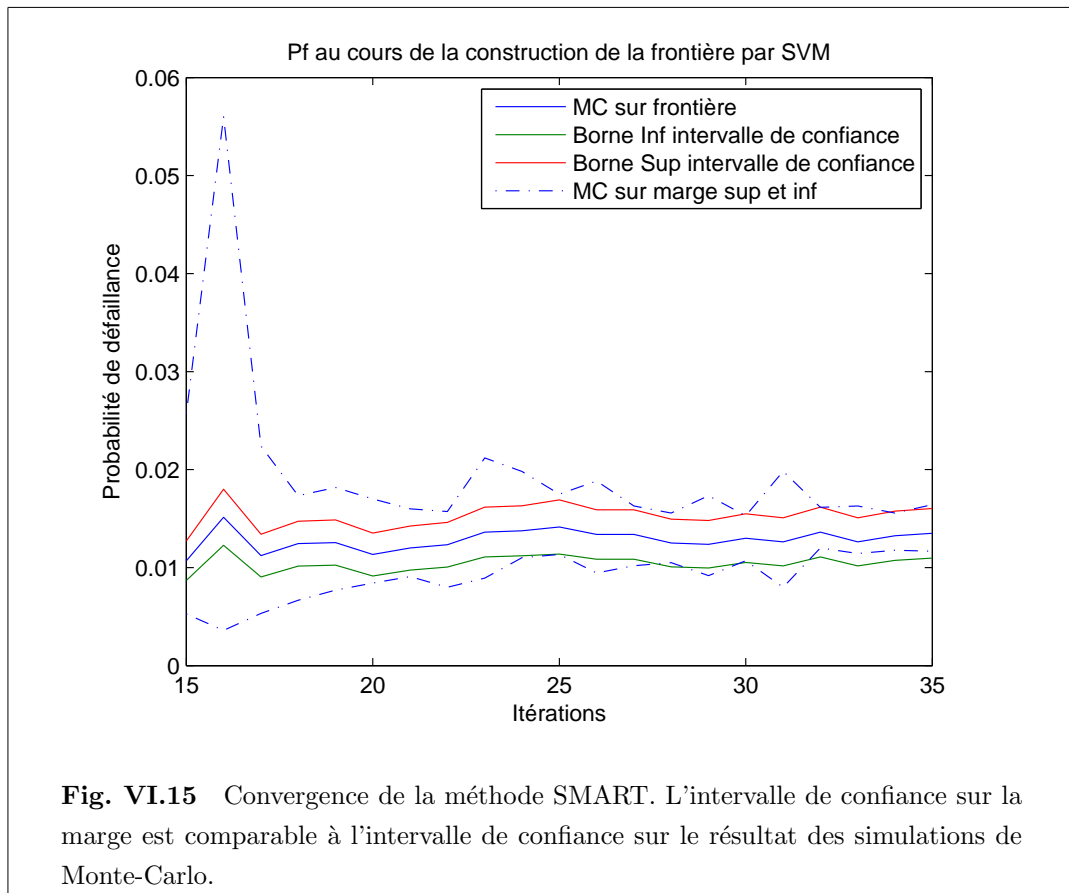
Tab. VI.8 Résultats fiabilistes obtenus pour la plaque complète.

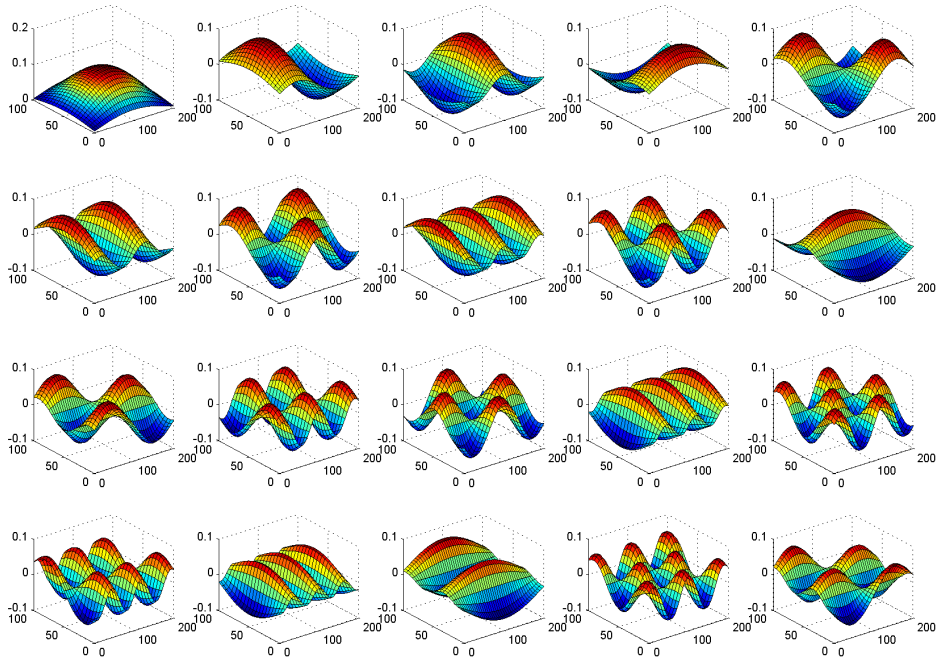


## Discussion

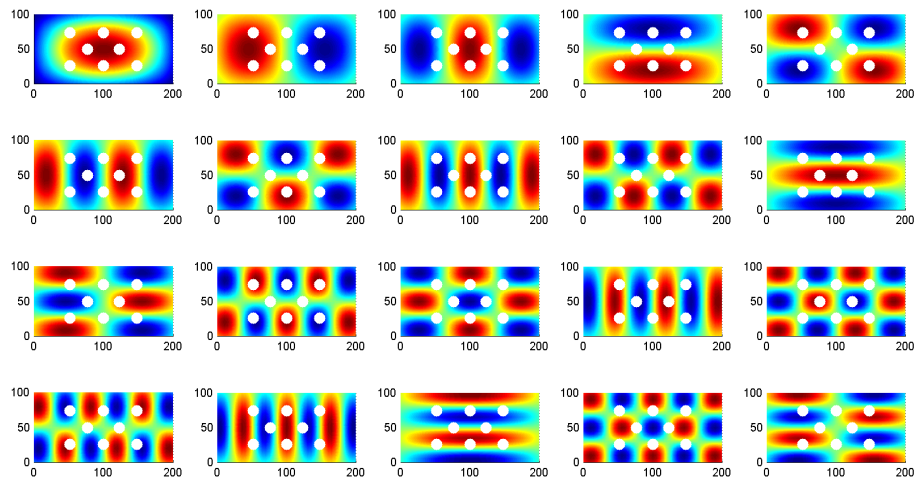
Les résultats obtenus par notre procédure sur cette application montrent une bonne représentation du domaine de défaillance par la surface SVM. La simulation sur la frontière analytique donne un résultat précis pour un nombre d'évaluations de l'état-limite qui est de l'ordre de la méthode FORM. L'apprentissage nécessite sur cet exemple un nombre d'appels supérieur à la méthode FORM, mais <sup>2</sup>SMART permet ici de trouver la probabilité de défaillance proche de la valeur de référence, obtenue ici par SS(AB)), contrairement à FORM.

Pour cette dernière application, la convergence de la simulation de Monte-Carlo sur la frontière SVM est présentée sur la figure VI.15. Un intervalle de confiance de la simulation de Monte-Carlo est défini à partir du coefficient de variation. La figure montre également la stabilité de la séparation des domaines en donnant des résultats de simulations réalisées sur chacun des deux côtés de la marge.





**Fig. VI.16** Modes du champ pour la plaque complète. À chaque variable aléatoire définissant le champ du module de Young est associé un mode du champ.



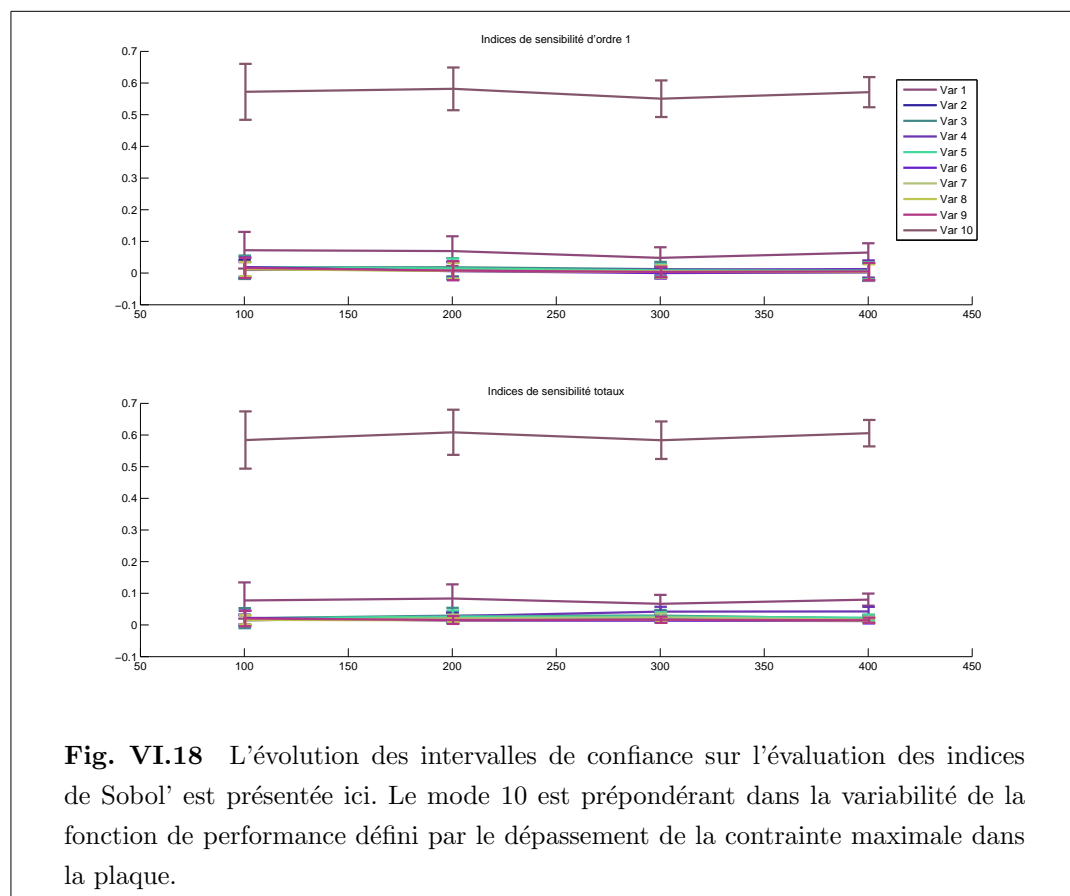
**Fig. VI.17** Cette figure représente les iso-valeurs associées à chaque variable définissant le champ du module de Young.

### Analyse de sensibilité

Une analyse de sensibilité est conduite sur le cas de la plaque complète. Il est nécessaire de représenter physiquement la valeur des 20 variables aléatoires. En fait, en décomposant le champ par la méthode EOLE, chaque variable est associée à un mode du champ aléatoire. La figure VI.16 présente le mode associé à chacune des variables, et la figure VI.17 la projection de ces modes sur la plaque trouée étudiée.

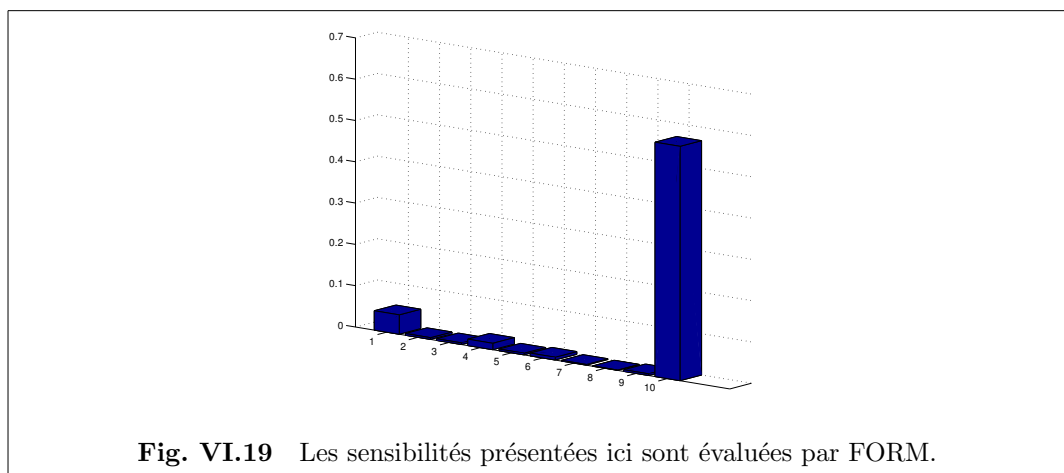
L'analyse de sensibilité va permettre de déterminer le mode prépondérant dans la défaillance.

Les résultats obtenus par simulation sur la formulation analytique de la fonction d'état-limite par régression SVM sont présentés figure VI.18. Deux modes sont prépondérants dans la défaillance : le mode 10 de manière évidente puis le mode 1. Le mode 1 influence le module de Young au centre de la plaque. Le mode 10 influence le module de Young également au centre de la plaque mais de manière encore plus localisée sur les deux trous au centre.



**Fig. VI.18** L'évolution des intervalles de confiance sur l'évaluation des indices de Sobol' est présentée ici. Le mode 10 est prépondérant dans la variabilité de la fonction de performance défini par le dépassement de la contrainte maximale dans la plaque.





En comparant ces deux modes prépondérants aux points de défaillance obtenus par l'analyse multi-FORM, nous obtenons une cohérence des approches en fiabilité et en sensibilité. Les indices de sensibilités FORM sont également cohérents avec cette analyse, cf. figure VI.19.

Il serait intéressant dans ce cas, afin d'éviter l'utilisation de l'approche multi-FORM, d'utiliser le *clustering* pour déterminer les points de défaillance à partir des simulations par *subsets*. En effet, les points obtenus lors du dernier pas de *subsets* permettraient d'utiliser l'approche présentée par F.Perrin dans [PPS06] pour les simulations directionnelles.

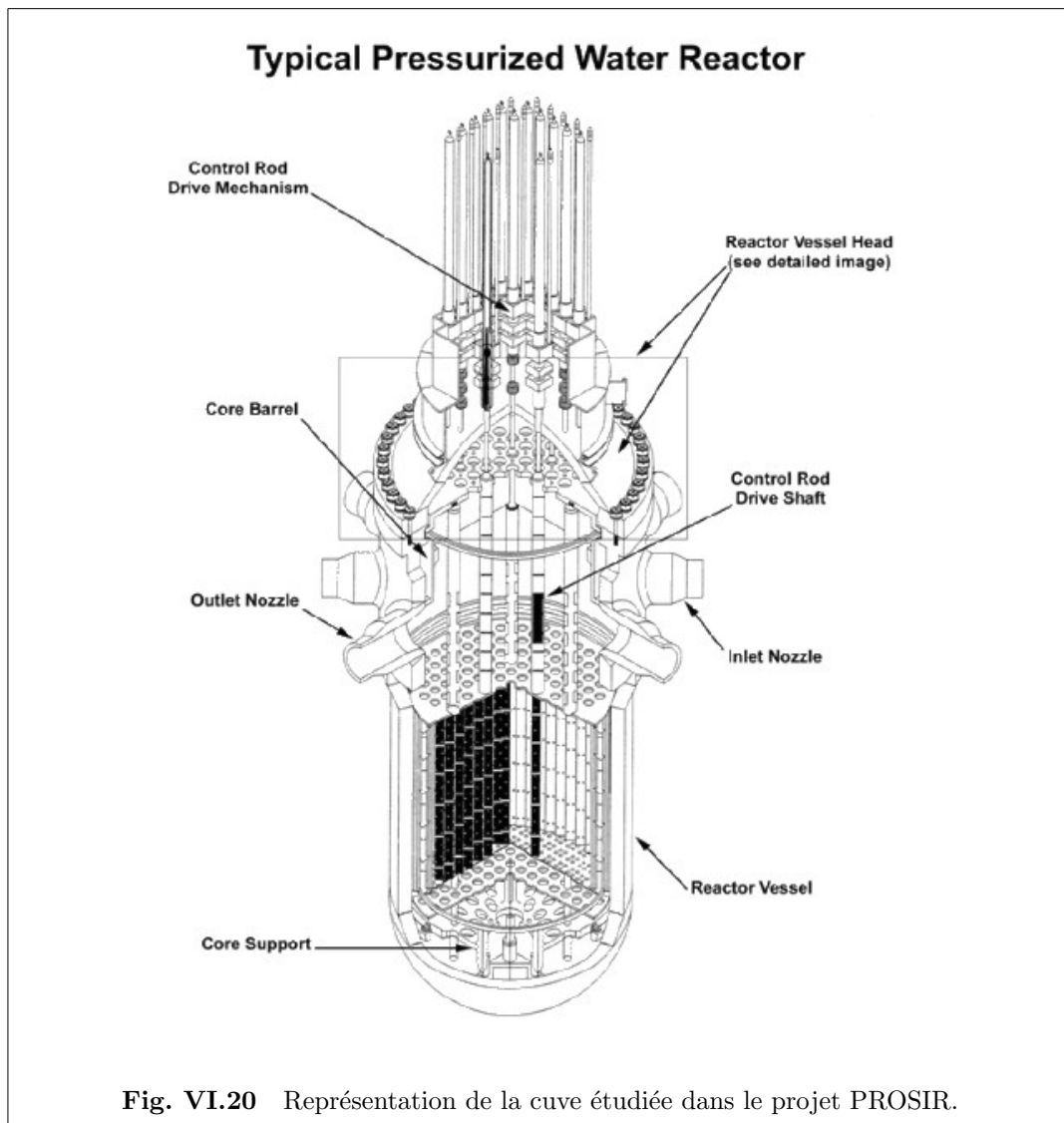
## VI.4 Le projet PROSIR

PROSIR, ou *Round Robins on Probabilistic Approach for Structural Integrity of Reactor Pressure Vessel* [PRO06].

Dans le cadre du comité de sécurité des installations nucléaires de l'OCDE-NEA, le groupe de travail sur l'intégrité des composants et structures est en charge de l'étude sur l'âge des composants et des structures des centrales nucléaires.

Une des clés réside dans l'étude des effets des radiations sur la cuve, dont un schéma est proposé en figure VI.20.

Les incertitudes prises en compte concernent les transitions thermiques, les propriétés matériaux, la distribution des défauts. L'approche déterministe est très conservatrice mais coûteuse et l'apport de l'approche probabiliste est examinée.



L'ensemble des données du problème (données cohérentes mais pas réelles...) sont fournies à différentes unités de recherche publiques et privées pour comparer méthodes et résultats. Voici la liste des participants :

- NRC-ORNL, USA
- AREVA, GmbH/Germany - ANP/France
- NRI, Czech
- KINS, Korea
- CEA, France
- JAEA, Japan

- EDF, France
- DNV, Sweden
- IE-JRC, EC
- TECNATOM, Spain

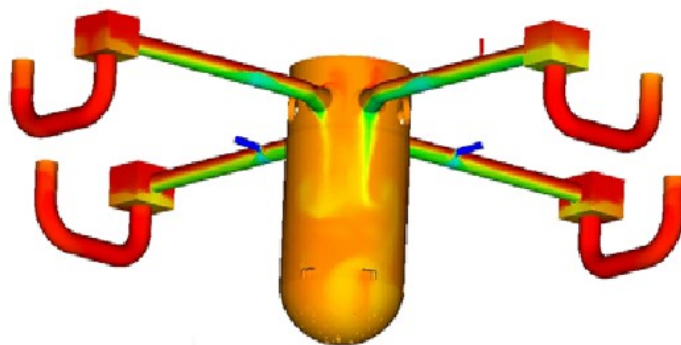
Une première phase du projet consiste à comparer les résultats sur l'approche déterministe, puis dans une seconde phase une approche probabiliste est mise en œuvre par les participants.

Ces données étant devenues libres d'exploitation, nous proposons notre approche pour résoudre ce problème de fiabilité. Deux points sont intéressants pour la validation en plus du simple fait que cette application soit industrielle. L'état-limite est considéré comme complexe de par la présence de points de conception multiples et les niveaux de fiabilité sont très importants, autrement dit, les probabilités de défaillance sont très faibles.

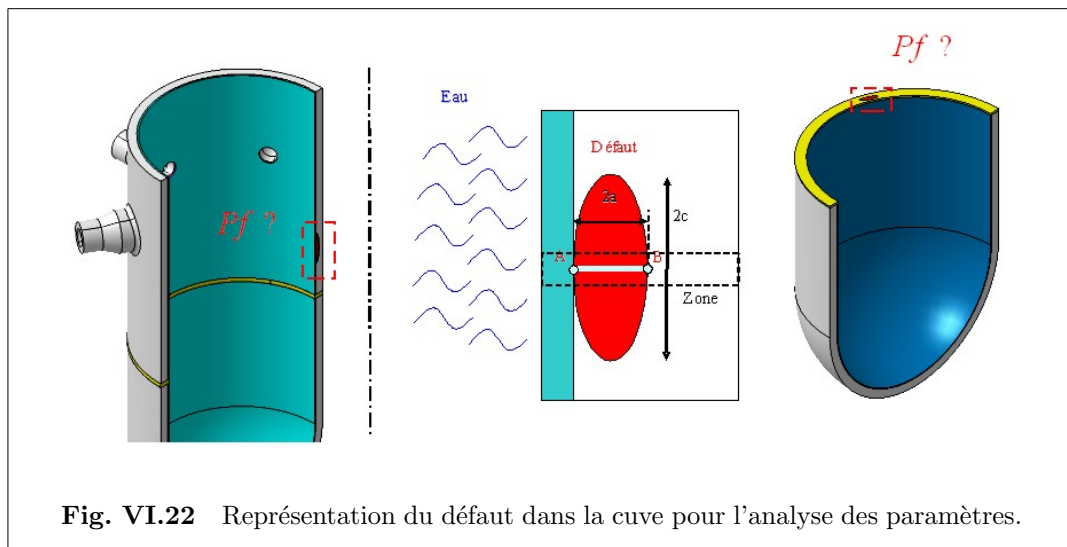
#### VI.4.1 Description du modèle mécanique

Il s'agit d'une étude thermo-mécanique de la résistance d'une cuve face à un choc thermique pressurisé suite à une injection d'eau de refroidissement. Le champ thermique en question est représenté figure VI.21.

Le modèle utilisé est un modèle unidimensionnel. Il s'agit de déterminer la nocivité d'un défaut (elliptique, circonférentiel ou longitudinal, figure VI.22) sous le revêtement, en tenant compte de l'effet de l'irradiation sur les propriétés à rupture des matériaux (métal de base et joint soudé).



**Fig. VI.21** Représentation d'un chargement thermique dans la cuve.



**Fig. VI.22** Représentation du défaut dans la cuve pour l'analyse des paramètres.

Les variables utilisées dans le modèle stochastique du projet PROSIR sont décrites dans le tableau VI.9.

Variables aléatoires	Loi	Moyenne	Écart-type
$U_{RTNDT}$	Normale	0	1
$U_{K1C}$	Normale	0	1
$A$ ( mm)	Déterministe	12	—
$C$ ( mm)	Déterministe	72	—
Âge (années)	Paramètre	10; 20; 40 ou 60	—
$RTNDT_{INIT}$	Normale	−20	9
Cuivre	Normale	0,086	0,01
Phosphore	Normale	0,0137	0,001
Nickel	Normale	0,72	0,05
Fluence	Normale	0	1

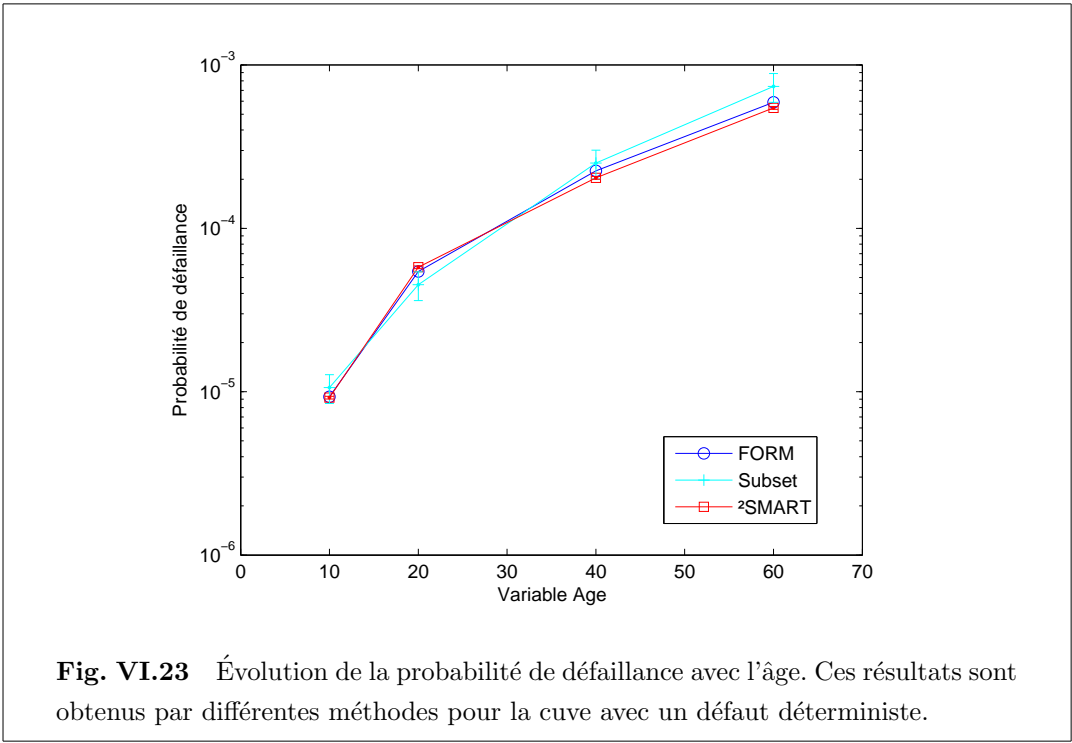
**Tab. VI.9** Variables aléatoires pour le projet PROSIR.

### VI.4.2 Résultats fiabilistes

Les tableaux VI.10 et VI.11 présentent les résultats obtenus par différentes méthodes de fiabilité. La référence considérée est le résultat obtenu par *subsets simulations* avec 2000 simulations par pas.

Âge (années)		FORM	SS(AB)	<sup>2</sup> SMART
10	$N$ appels	200	10000	1500
	$P_f$ ( $10^{-5}$ )	0,98	1,12	0,92
	$\beta$	4,27	4,24	4,28
20	$N$ appels	200	10000	1200
	$P_f$ ( $10^{-5}$ )	5,75	8,67	5,78
	$\beta$	3,86	3,75	3,86
40	$N$ appels	200	8000	1200
	$P_f$ ( $10^{-4}$ )	2,38	2,23	2,04
	$\beta$	3,49	3,51	3,53
60	$N$ appels	200	8000	1200
	$P_f$ ( $10^{-4}$ )	6,35	6,05	5,47
	$\beta$	3,22	3,24	3,27

**Tab. VI.10** Résultats fiabilistes obtenus pour le projet PROSIR. Le défaut est ici déterministe.



**Fig. VI.23** Évolution de la probabilité de défaillance avec l'âge. Ces résultats sont obtenus par différentes méthodes pour la cuve avec un défaut déterministe.

Les différentes méthodes employées donnent des résultats plus ou moins équiva-

lents. La figure VI.23 présente l'évolution de la fiabilité en fonction de l'âge de la cuve. Lorsque les caractéristiques du défaut sont déterministes, la méthode FORM est la plus performante sur ce calcul.

### VI.4.3 Le défaut aléatoire

Le modèle stochastique décrit dans le tableau VI.9 contient une simplification importante. La géométrie du défaut est déterministe. Pour plus de réalisme, un aléa est apporté à la description du défaut. Les variables  $A$  et  $C$  sont probabilisées, mais restent liées :  $C = 6A$ .

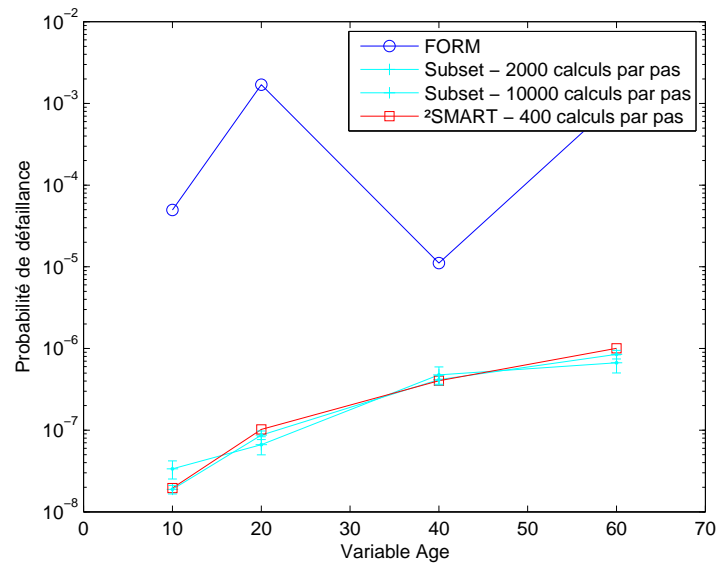
La probabilité sur  $A$  est décrite comme suit :

$$\begin{cases} P(A < x) = 1 - \exp(-689x) \text{ pour } x < 8,559 \text{ mm} \\ P(A < x) = 1 - 0,02 \exp(-232x) \text{ pour } 8,559 \text{ mm} \leq x \end{cases} \quad (\text{VI.7})$$

La variabilité introduite dans le défaut perturbe de manière importante la résolution et les résultats du calcul. La probabilité de défaillance diminue nécessairement puisqu'on ajoute de la variabilité. Mais cet aléa influence fortement la convergence de la méthode FORM, méthode qui devient inutilisable dans ce cas.

Âge (années)		FORM	SS(AB)	SS(AB)	<sup>2</sup> SMART
10	$N$ appels	200	16000	80000	4500
	$P_f$ ( $10^{-8}$ )	4960	3,36	1,89	1,96
	$\beta$	3,89	5,40	5,50	5,49
20	$N$ appels	200	16000	80000	4500
	$P_f$ ( $10^{-8}$ )	170000	6,66	8,71	10,23
	$\beta$	2,93	5,27	5,22	5,20
40	$N$ appels	200	14000	70000	4000
	$P_f$ ( $10^{-7}$ )	111	4,75	4,15	4,02
	$\beta$	4,24	4,90	4,93	4,93
60	$N$ appels	200	14000	70000	4000
	$P_f$ ( $10^{-7}$ )	10600	6,70	8,47	10,07
	$\beta$	3,07	4,83	4,79	4,75

**Tab. VI.11** Résultats fiabilistes obtenus pour le projet PROSIR avec le défaut aléatoire.



**Fig. VI.24** Évolution de la probabilité de défaillance avec l'âge. Ces résultats sont obtenus par différentes méthodes pour la cuve avec un défaut aléatoire.

## VI.5 Conclusion

Ce dernier chapitre a permis l'illustration des méthodes SMART et <sup>2</sup>SMART sur des cas issus de la mécanique. La régression SVM pour l'étude de la sensibilité des variables d'entrée a également été testée. Ces deux approches, basées sur l'apprentissage par les SVM, soit de la frontière par classification pour la fiabilité, soit de la fonction de performance pour la sensibilité, ont montré une complémentarité, et une réelle efficacité.

## Conclusions et perspectives

**P**RENDRE en compte l'incertain probabiliste en conception mécanique a conduit au développement de deux types d'analyses : d'une part l'analyse de fiabilité qui recherche les conditions de la défaillance en vue d'assurer un dimensionnement optimal et d'autre part l'analyse de sensibilité qui recherche l'influence des paramètres pour une conception robuste.

Au-delà des cas académiques, la mise en œuvre de telles analyses se heurte à des difficultés liées à la complexité des calculs mécaniques et numériques cherchant à apporter une représentation de plus en plus fine de la physique.

S'appuyant sur la longue expérience du partenariat entre l'IFMA, EdF et Phiméca et sur un acquis important du laboratoire, l'objectif de notre recherche visait à introduire les méthodes de l'apprentissage statistique afin de construire des substituts aux modèles mécaniques, à définir les conditions de mise en œuvre de ces méthodes et à valider leur utilisation.

L'analyse de la littérature, visant à faire le bilan des méthodes existantes, a permis de dégager une piste de travail répondant à nos objectifs : la substitution de l'état-limite. Cette voie semblait la plus efficace par rapport aux critères d'augmentation du nombre de variables et de complexité des modèles. De plus, la substitution de l'état-limite permet de conserver les méthodes qui ont fait leurs preuves aujourd'hui : les simulations ou les méthodes analytiques. Ce type d'approche permet également de raisonner en terme de base de données et de construire le modèle de substitution à l'aide de différents outils conservant une approche totalement non intrusive par rapport au modèle numérique ou analytique de l'étude.

L'exigence de souplesse de l'approximation, nécessaire, a orienté notre travail vers les outils de l'apprentissage statistique dont sont issus les méthodes à noyaux,



les réseaux de neurones et les machines à vecteurs supports. Les SVM, dont l'intérêt pour la fiabilité a fait l'objet du livre de J.E. Hurtado [Hur04b], ont particulièrement retenu notre attention.

La théorie des SVM vise à une classification, ou une régression, optimale par rapport à un ensemble de points donnés. Rien n'y est dit sur la stratégie optimale d'ajout de points qui a été l'un des objectifs de la thèse. Les développements SVM sont essentiellement orientés vers le traitement de grandes bases de données, mais ils possèdent aussi une forte capacité de généralisation. Cette dernière propriété est particulièrement attrayante pour traiter des problèmes de fiabilité où la taille des échantillons est très faible.

Le cœur du travail de thèse a été de définir une stratégie optimale de construction de la base de données pour décrire l'état-limite, c'est-à-dire la frontière entre les domaines de sûreté et de défaillance. En nous basant sur l'algorithme SVM pour la construction d'un séparateur, nous avons adapté une stratégie d'apprentissage pour la fiabilité. Il s'agissait de minimiser le nombre de points d'apprentissage et le nombre d'itérations du processus de construction du séparateur optimal pour économiser les coûts de calcul. Différents critères ont été créés à partir des informations issues de la marge et de son évolution au cours des itérations afin de suivre le processus de construction. Ces critères ont permis de proposer la méthode *SMART : Support vector Margin Algorithm for Reliability esTimation*. Elle permet de construire de manière itérative l'état-limite par la sélection aléatoire de points répartis de manière homogène dans la marge et de points considérés instables. Une simulation de Monte-Carlo sur l'état-limite analytique permet ensuite d'obtenir la probabilité de défaillance.

Confrontés à un problème de sélection de paramètres au démarrage de l'algorithme, paramètres liés à la valeur de la probabilité de défaillance finale (par exemple le nombre de simulations nécessaires à l'évaluation de la probabilité de défaillance), nous avons proposé de coupler SMART à la méthode des *subsets simulations*. La classification et l'apprentissage s'effectuent sur des densités conditionnées, donc limitées dans l'espace et non plus sur l'espace complet de définition de la fonction de performance. Ce couplage efficace est intégré dans la méthode <sup>2</sup>*SMART : Subsets by Support vector Margin Algorithm for Reliability esTimation*.

Les approches proposées ont été validées sur des cas tests de la littérature. Ces états-limites mathématiques ont l'avantage de présenter des difficultés rencontrées lors des études de fiabilité et donc de confronter les méthodes proposées à la prise en compte de la dimension, de la forme complexe de la fonction de performance et

des points de conception multiples. La progression de l'apprentissage s'est avérée satisfaisante. Après l'étude des cas tests, nous soulignons les points suivants :

- les points d'apprentissage se concentrent bien dans la marge et autour des zones (et non de la zone) où la densité de probabilité est la plus forte ;
- les valeurs de la probabilité de défaillance obtenues sont sans biais et la dispersion est relativement faible ;
- l'incertitude sur la marge est incluse dans l'intervalle de confiance obtenu par l'évaluation de la probabilité par simulation ;
- le cumul des approximations faites à chaque pas des *subsets* ne pénalise pas la valeur de la probabilité estimée.

Les résultats satisfaisants obtenus sur l'analyse de fiabilité, et l'utilisation de la régression SVM a naturellement conduit notre curiosité vers l'analyse de sensibilité. En effet, le plan initial utilisé dans <sup>2</sup>SMART est un plan aléatoire dans l'espace standard. Ce type de plan est utilisé pour la construction de surfaces de réponse. Nous utilisons alors cette information pour en tirer des sensibilités globales de type indices de Sobol', sensibilités mesurées par l'analyse de la variance. Associée à une méthode de *resampling*, la souplesse de l'approximation de la fonction de performance par SVM a donné des résultats satisfaisants.

Contrairement à ce qui est fait pour la fiabilité, le plan d'expériences n'est pas optimisé pour la régression. Le plan aléatoire est satisfaisant mais il ne permet pas de créer de manière itérative un plan spécialement conçu par rapport à des critères d'erreur. Un début de réponse est apporté par les *sparse grid*. Les points de la grille sont ajoutés en fonction des dimensions influentes ou celles dans lesquelles l'erreur est la plus importante. Les premiers essais sont encourageants et mériteront un approfondissement.

Les méthodes proposées ont finalement été testées sur des applications de la mécanique. Elles ont été choisies pour les difficultés de résolution qu'elles présentent avec les méthodes classiques. Les approches proposées, basées sur l'utilisation des SVM en fiabilité ou en sensibilité, ont montré une efficacité certaine et une complémentarité.

Ce travail ouvre des perspectives de plusieurs ordres.

En premier lieu, la validation de la méthode <sup>2</sup>SMART sur des cas réels et en nombre plus important doit être poursuivie. Ce travail pourra être réalisé par l'intégration des méthodes développées dans le logiciel Phimeca Software et dans une

contribution au projet ANR CISSSI.

En second lieu, la partie sensibilité, abordée dans le chap V, nécessite d'être approfondie. Il s'agit notamment d'optimiser la construction du plan d'expériences en essayant d'adapter les nouvelles expériences aux dimensions importantes et aux zones où l'erreur mesurée nécessite une amélioration de l'approximation. Pour cette partie analyse de sensibilité, une phase de tests permettraient également d'améliorer les performances.

La recherche des points de conception n'a pas fait l'objet d'un travail dans cette thèse, y compris dans la partie fiabilité puisque notre objectif était le calcul de la probabilité de défaillance par simulations. Cependant cette information est très utile. L'utilisation de *clustering* pour leur détermination après des simulations ouvre une voie. L'information sur les sensibilités locales est un complément aux sensibilités globales de type indices de Sobol'.

Enfin, ce travail nous a montré l'efficacité de croiser les compétences des différents domaines (les SVM étant issus des mathématiques et étant largement utilisés en vision ou en génétique...). L'une des perspectives sera donc de continuer un travail de veille sur des outils qui font leurs preuves dans des domaines où nous avons à apprendre, pour les adapter à nos objectifs.

# Bibliographie

- [AB01] S.K. Au and J.L. Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16, 2001.
- [AR64] E. Aizerman, M. Braverman and L. Rozonoerk. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.
- [ARRZ03] D. Anguita, S. Ridella, F. Riveccio, and R. Zunino. Hyperparameter design criteria for support vector classifiers. *Neurocomputing*, 55:109–134, 2003.
- [BA05] J.L. Beck and S.K. Au. Reliability of dynamic systems using stochastic simulation. In *EURODYN 2005*, 2005.
- [Ber05] M. Berveiller. *Éléments finis stochastiques : approches intrusive et non intrusive pour des analyses de fiabilité*. PhD thesis, Université Blaise Pascal, Clermont-Ferrand, 2005.
- [BG04a] Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5:1089–1105, 2004.
- [BG04b] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, (13):147–269, 2004.
- [BGV92] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *5th Annual ACM Workshop on COLT*, 1992.
- [BH89] K. Breitung and M. Hohenbichler. Asymptotic approximations for multivariate integrals with an application to multinormal probabilities. *Journal of Multivariate Analysis*, 30(1):80–97, 1989.

- [Bla06] G. Blatman. Méthodes non intrusives pour les analyses de sensibilité en modélisation probabiliste. Master's thesis, Université Blaise Pascal, Clermont-Ferrand, 2006.
- [BM02] S. Bandyopadhyay and U. Maulik. An evolutionary technique based on K-means algorithm for optimal clustering in  $\mathbb{R}^n$ . *Information Sciences*, 146(1-4):221–237, 2002.
- [BNR00] V. Barthelmann, E. Novak, and K. Ritter. High dimensional polynomial interpolation on sparse grids. *Adv. Comput. Math.*, 4(12):273–288, 2000.
- [CAD99] CADOE. Variational analysis using high order derivatives. Technical report, CADOE, 1999.
- [CD00] K.K. Choi and W. Duan. Design sensitivity analysis and shape optimization of structural components with hyperelastic material. *Computer methods in applied mechanics and engineering*, 187:219–243, 2000.
- [CGS03] S.M. Clarke, J.H. Griebisch, and T.W. Simpson. Analysis of support vector regression for approximation of complex engineering analyses. In *The 2003 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, 2003.
- [Cha02] O. Chapelle. *Support Vector Machine: Principes d'induction, Réglage automatique et Connaissances a priori*. PhD thesis, Université Paris VI, 2002.
- [CM03a] P. Cusdin and J.D. Muller. Automatic differentiation : learning to speak ad. Technical report, School of Aeronautical Engineering, Queen's University, Belfast, 2003.
- [CM03b] P. Cusdin and J.D. Muller. Automatic differentiation and sensitivity analysis methods for cfd. Technical report, School of Aeronautical Engineering, Queen's University, Belfast, 2003.
- [Cor92] G. Corliss. Automatic differentiation bibliography. Technical report, Center for research on Parallel Computation, Rice University, Houston, 1992.
- [CST00] N. Cristianni and J. Shaw-Taylor. *An Introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.

- [CVBM02] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159, 2002.
- [DGLQY05] J. Deng, D. Gu, X. Li, and Z. Qi Yue. Structural reliability analysis for implicit performance functions using artificial neural network. *Structural Safety*, 27:25–48, 2005.
- [DKD98] A. Der Kiureghian and T. Dakessian. Multiple design points in first and second-order reliability. *Structural Safety*, 20:37–49, 1998.
- [DKP86] A. Der Kiureghian and Liu P.L. Structural reliability under incomplete probability information. *Journal of Engineering Mechanics ASCE*, 112(1):85–104, 1986.
- [DL06] F. Deheeger and M. Lemaire. Reliability analysis by support vector machine classification. In *Das P.K. ed. - Integrating Structural Analysis, Risk & Reliability, 3rd ASRANet colloquium - Glasgow*, 2006.
- [DL07] F. Deheeger and M. Lemaire. Support vector machine for efficient subsets simulation: <sup>2</sup>SMART method. In *Kanda J., Takada T. and Furuta H. eds. - Applications of Statistics and Probability in Civil Engineering, 10th ICASP - Tokyo*, 2007.
- [DM05] O. Ditlevsen and H.O. Madsen. *Structural Reliability Methods*. John Wiley, 2005.
- [DSDK90] M. De Stefano and A. Der Kiureghian. An efficient algorithm for second-order reliability analysis. report no. ucb/sem-90/20. Technical report, Division of Structural Engineering and Structural Mechanics, Department of Civil Engineering, University of California, Berkeley, 1990.
- [Ele05] C. Elegbede. Structural reliability assessment based on particles swarm optimization. *Structural Safety*, 27:171–186, 2005.
- [ET93] B. Efron and R. Tibshirani. *An introduction to the Bootstrap*. Chapman and Hall, New York, 1993.
- [FC04] J. Fortuna and D. Capson. Improved support vector classification using pca and ica feature space modification. *Pattern recognition*, 37:1117–1129, 2004.
- [FER] FERUM. [www.ce.berkeley.edu/FERUM](http://www.ce.berkeley.edu/FERUM).
- [Fin99] T. Fine. *Feedforward Neural Network Methodology*. Springer, 1999.

- [GA04] H.M. Gomes and A.M. Awruch. Comparison of response surface and neural network with other methods for structural reliability analysis. *Structural Safety*, 26:49–67, 2004.
- [GBL03] N. Gayton, J.M. Bourinet, and M. Lemaire. CQ2RS: a new statistical approach to the response surface method for reliability analysis. *Structural Safety*, 25(4):99–121, 2003.
- [GG03] T. Gernster and M. Griebel. Dimension-adaptative tensor-product quadrature. *Computing*, 1(71):65–87, 2003.
- [GH00] P. Guillaume and A. Huard. Multivariate Padé approximation. *Journal of computational and applied mathematics*, 121:197–219, 2000.
- [GHR97] P. Guillaume, A. Huard, and V. Robin. Generalized multivariate Padé approximants. *Journal of Approximation Theory*, 95:203–214, 1997.
- [GM94] P. Guillaume and M. Masmoudi. Computation of high order derivatives in optimal shape design. *Numer. Math.*, 67:231–250, 1994.
- [GM01] X.L. Guan and R.E. Melchers. Effect of response surface parameter variation on structural reliability estimates. *Structural Safety*, 23(4):429–444, 2001.
- [GS91] R.G. Ghanem and P.D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer, Berlin, 1991.
- [Gui97a] P. Guillaume. Convergence of the nested multivariate Padé approximants. *Journal of Approximation Theory*, 94:455–466, 1997.
- [Gui97b] P. Guillaume. Nested multivariate Padé approximants. *Journal of computational and applied mathematics*, 82:149–158, 1997.
- [Gun98] S.V. Gunn. Support vector machines for classification and regression. Technical report, Department of Electronics and Computer Science, University of Southampton, 1998.
- [HA01] J.E. Hurtado and D.A. Alvarez. Neural-network-based reliability analysis: a comparative study. *Computer Methods in Applied Mechanics and Engineering*, 191:113–132, 2001.
- [HA03] J.E. Hurtado and D.A. Alvarez. A classification approach for reliability analysis with stochastic finite element modeling. *Journal of Structural Engineering*, 129:1141–1149, 2003.

- [Has70] W.K. Hastings. Monte-Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [Hau03] T. Haukaas. *Finite Elements Reliability and Sensitivity Methods for Performance Based Engineering*. PhD thesis, University of California, Berkeley, 2003.
- [HCL98] C.W. Hsu, C.C. Chang, and C.J. Lin. A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering National Taiwan University, 1998.
- [HS96] T. Homma and A. Saltelli. Importance measures in global sensitivity analysis of non linear models. *Reliability Engineering and System Safety*, 52:1–17, 1996.
- [Hur04a] J.E. Hurtado. An examination of methods for approximating implicit limit state functions from the viewpoint of statistical learning theory. *Structural Safety*, 26:271–293, 2004.
- [Hur04b] J.E. Hurtado. *Structural reliability, Statistical Learning Perspectives*. Lecture Notes in Applied and Computational Mechanics Vol.17, Springer editions, 2004.
- [Jac05] J. Jacques. *Contributions à l’analyse de sensibilité et à l’analyse discriminante généralisée*. PhD thesis, Université Joseph Fourier, Grenoble, 2005.
- [Jul97] L.S. Julian. *Resampling: The New Statistics - Second Edition published October 1997*. <http://www.resample.com>, 1997.
- [KH02] S.H. Kim and H. Hub. Design sensitivity analysis of sheet metal forming processes with a direct differentiation method. *Journal of materials processing technology*, 130:504–510, 2002.
- [Kli06a] A. Klimke. Sparse Grid Interpolation Toolbox – user’s guide. Technical Report IANS report 2006/001, University of Stuttgart, 2006.
- [Kli06b] A. Klimke. *Uncertainty modeling using fuzzy arithmetic and sparse grids*. PhD thesis, Universität Stuttgart, Shaker Verlag, Aachen, 2006.
- [KM04] I. Kaymaz and C.A. McMahon. A response surface method based on weighted regression for structural reliability analysis. *Probabilistic Engineering Mechanics*, 2004.



- [KW05] A. Klimke and B. Wohlmuth. Algorithm 847: spinterp: Piecewise multilinear hierarchical sparse grid interpolation in MATLAB. *ACM Transactions on Mathematical Software*, 31(4), 2005.
- [LADK86] P-L. Liu and A. A. Der Kiureghian. Optimization algorithms for structural reliability analysis. report no. ucb/sesm-86/09. Technical report, Division of Structural Engineering and Structural Mechanics, Department of Civil Engineering, University of California, Berkeley, 1986.
- [Lem04] M. Lemaire. How to manage industrial studies. *Det Norske Veritas*, 2004.
- [Lem05] M. Lemaire. *Fiabilité des structures - Couplage mécano-fiabiliste statique*. Hermes Science, 2005.
- [Lem08] M. Lemaire. *Structural Reliability*. ISTE - to appear, 2008.
- [LWS05] X. Li, L. Wang, and E. Sung. Adaboost with svm-based component classifiers. *IEEE Transactions on System, Man and Cybernetics, PartB: Cybernetics*, 2005.
- [MAT03] MATLAB. [www.mathworks.com](http://www.mathworks.com), 2003.
- [Men99] J.B. Menzies. What researchers do and what practitioners need. *Structural Safety*, 21:349–356, 1999.
- [MKL86] H.O. Madsen, S. Krenk, and N.C. Lind. *Methods of Structural Safety*. Prentice-Hall, 1986.
- [MRR<sup>+</sup>53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- [NIS06] NIST. *NIST/SEMATECH e-Handbook of Statistical Methods*. <http://www.itl.nist.gov/>, 2006.
- [Nor06] E. Noret. Approche stochastique du dimensionnement des structures et systèmes. Master's thesis, IFMA, Clermont-Ferrand, 2006.
- [NQR98] D.T. Nguyen, R. Qamar, and H. Runesha. Automatic differentiation for design sensitivity analysis of structural systems using parallel vector processors. *Advances in engineering software*, 29:375–382, 1998.
- [Pen02] P.C. Pendharkar. A computational study on the performance of artificial neural networks under changing structural design and data distribution. *European Journal of Operational Research*, 138:155–177, 2002.

- [PHI06] PHIMECA Engineering SA. *Phimeca Software - User's manual*, 2006.
- [PL02] M. Papadrakakis and N.D. Lagaros. Reliability-based structural optimization using neural networks and monte-Carlo simulation. *Computer Methods in Applied Mechanics and Engineering*, 32:3491–3507, 2002.
- [PPS06] F. Perrin, M. Pendola, and F. Suau. Design points in directional simulation. In *ASRANet Third International Colloquium 2006 - Glasgow*, 2006.
- [PRO06] PROSIR. Probabilistic structural integrity of a PWR reactor pressure vessel. In *Round Robins on Probabilistic Approach for Structural Integrity of Reactor Pressure Vessel, OECD - NEA PROSIR Workshop, Lyon, France*, 2006.
- [Rac01] R. Rackwitz. Reliability analysis - a review and some perspectives. *Structural Safety*, 23:365–395, 2001.
- [RB03] D. Roos and C. Bucher. Adaptive response surfaces for structural reliability of nonlinear finite element structures. In *NAFEMS Seminar*, 2003.
- [Rob96] C.P. Robert. *Méthodes de Monte Carlo par chaînes de Markov*. Economica, 1996.
- [Sap90] G. Saporta. *Probabilités, analyse des données et statistique*. éditions Technip, 1990.
- [SBL04] B. Sudret, M. Berveiller, and M. Lemaire. Eléments finis stochastiques en élasticité linéaire. *Comptes Rendus Mécanique*, 332:531–537, 2004.
- [Sch00] B. Schölkopf. Statistical learning and kernel methods. Technical report, Microsoft research limited, 2000.
- [SCS00] A. Saltelli, K. Chan, and E.M. Scott. *Sensitivity analysis*. Wiley, 2000.
- [SDK00] B. Sudret and A. Der Kiureghian. Stochastic finite element methods and reliability, a state-of-the-art report, report no. ucb/sem-2000/08. Technical report, Department of Civil and Environmental Engineering, University of California, Berkeley, 2000.
- [SF96] R. Serban and J.S. Freeman. Direct differentiation method for the design sensitivity of multibody dynamics system. In *The 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, 1996.

- [Sob93] I.M. Sobol'. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiments*, 1:407–414, 1993.
- [SPI06] SPIDER. [www.kyb.tuebingen.mpg.de/bs/people/spider](http://www.kyb.tuebingen.mpg.de/bs/people/spider), 2006.
- [SS03a] B. Schölkopf and A. Smola. Kernel methods and support vector machines. Technical report, Max-Planck-Institut für biologische Kybernetik, 2003.
- [SS03b] A. Smola and B. Schölkopf. A tutorial on support vector regression. Technical report, 2003.
- [Sud07] B. Sudret. Global sensitivity analysis using polynomial chaos expansion. *Reliability Engineering & System Safety*, Article in press, 2007.
- [TBBA99] B. Thon, D. Bariant, S. Bila, and Al. Coupled Padé approximation-finite element method applied to microwave device design. Technical report, CADOE, 1999.
- [VA71] V. Vapnik and Chervonenkis A. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.
- [Vap98] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.
- [VL63] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24, 1963.
- [Waa00] P.H. Waarts. *Structural Reliability using Finite Element Method, An appraisal of DARS*. Delft University Press, 2000.
- [Wik06] Wikipedia. *Kernel trick* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/>, 2006.
- [Wik07a] Wikipedia. *AdaBoost* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/>, 2007.
- [Wik07b] Wikipedia. *Analysis of variance* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/>, 2007.
- [Wik07c] Wikipedia. *Latin hypercube sampling* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/>, 2007.
- [Wik07d] Wikipedia. *Padé approximant* — *Wikipédia, The Free Encyclopedia*. <http://en.wikipedia.org/>, 2007.

# Table des figures

I.1	Un portique . . . . .	8
I.2	Collaboration dans l'élaboration d'une étude de fiabilité . . . . .	9
I.3	Illustration des approximations FORM et SORM . . . . .	14
I.4	Illustration des simulations de Monte-Carlo . . . . .	23
I.5	Illustration des simulations d'importance . . . . .	24
I.6	Illustration des simulations directionnelles . . . . .	25
I.7	Illustration des simulations hypercube latin. . . . .	26
II.1	SVM et classification linéaire . . . . .	40
II.2	Le <i>Kernel Trick</i> . . . . .	44
II.3	Introduction des variables molles . . . . .	46
II.4	Fonctions d'erreur en régression . . . . .	47
II.5	Illustration de la complexité du méta-modèle . . . . .	52
II.6	Erreur d'apprentissage et erreur de généralisation . . . . .	54
II.7	VC dimension pour la droite dans le plan . . . . .	55
II.8	Illustration du principe SRM . . . . .	57
II.9	Paramètre $\sigma$ pour les fonctions RBF . . . . .	60
II.10	Compression par Quantification Vectorielle . . . . .	63
II.11	Quantification et régions de Voronoï . . . . .	65
III.1	L'exemple conducteur du chapitre . . . . .	72
III.2	Points de conception de l'exemple conducteur . . . . .	73
III.3	Algo de J.E. Hurtado - un premier séparateur . . . . .	75
III.4	Algo de J.E. Hurtado - évolution du séparateur . . . . .	77
III.5	Plan initial, choix de J.E. Hurtado . . . . .	80
III.6	Plan initial, carré latin . . . . .	81
III.7	Sélection efficace des points de la marge . . . . .	84

III.8	Utilité de la phase finale sur une population gaussienne . . . . .	86
III.9	Critères de validation - Nombre de points dans la marge . . . . .	89
III.10	Critères de validation - Superposition des marges successives . . . . .	89
III.11	Critères de validation - Stabilité de la classe des points . . . . .	90
III.12	Probabilité de défaillance au cours de l'apprentissage . . . . .	91
III.13	Évaluations de l'état-limite des points d'apprentissage . . . . .	92
III.14	Évolution des paramètres de la SVM . . . . .	93
III.15	Illustration de la décomposition en domaines. . . . .	99
III.16	<i>Subsets simulations</i> - Étape 2 . . . . .	100
III.17	<i>Subsets simulations</i> - Étape 3 . . . . .	101
III.18	<i>Subsets simulations</i> - Étape 4 . . . . .	101
III.19	<i>Subsets simulations</i> - Étape 5 . . . . .	102
III.20	<i>Subsets simulations</i> - Étape 6 . . . . .	102
III.21	Choix de la probabilité cible. . . . .	103
III.22	Organigramme <sup>2</sup> SMART : détermination du premier seuil . . . . .	105
III.23	Organigramme <sup>2</sup> SMART : stratégie d'apprentissage . . . . .	108
III.24	Illustration de <sup>2</sup> SMART sur l'exemple conducteur . . . . .	110
IV.1	Règle du nombre de points ajoutés pour l'apprentissage . . . . .	123
IV.2	Représentation graphique du séparateur - Cas $R - S$ . . . . .	124
IV.3	Suivi de convergence - Surface bruitée . . . . .	126
IV.4	Suivi de convergence - Surface bruitée . . . . .	126
IV.5	Suivi de convergence - la sphéroïde . . . . .	127
IV.6	Représentation graphique du séparateur - Cas convexe . . . . .	128
IV.7	Représentation graphique du séparateur - Cas concave . . . . .	129
IV.8	Illustration du séparateur - l'hyperbole . . . . .	131
IV.9	Illustration du séparateur - cas système en 2D . . . . .	132
V.1	Comparaison de générateurs aléatoires . . . . .	147
V.2	Schéma de la poutre sur 2 appuis . . . . .	149
V.3	Validation de la régression - Cas poutre en flexion . . . . .	150
V.4	Validation de la régression - Cas poutre en flexion . . . . .	151
V.5	Treillis à 23 barres . . . . .	152
V.6	Validation de la régression - Le treillis - Moments statistiques . . . . .	153
V.7	Validation de la régression - Le treillis - Indices de Sobol' . . . . .	154
V.8	Représentation de type de sparse grid . . . . .	156

V.9	Comparaison interpolation sparse grid et SVM . . . . .	158
V.10	Comparaison interpolation sparse grid et SVM . . . . .	159
VI.1	Sphère épaisse sous pression . . . . .	164
VI.2	Convergence de <sup>2</sup> SMART pour la sphère . . . . .	166
VI.3	Indices de Sobol' - Sphère sous pression . . . . .	167
VI.4	Sensibilités FORM - Sphère sous pression . . . . .	167
VI.5	Convergence de <sup>2</sup> SMART . . . . .	169
VI.6	Schéma de l'oscillateur . . . . .	169
VI.7	Convergence de <sup>2</sup> SMART . . . . .	171
VI.8	Indices de Sobol' - Le cas de l'oscillateur . . . . .	172
VI.9	Sensibilités FORM - Le cas de l'oscillateur . . . . .	173
VI.10	Résultats fiabilistes - Le cas de l'oscillateur . . . . .	174
VI.11	Discretisation par éléments finis de la plaque trouée. . . . .	175
VI.12	Réalisation du champ aléatoire $E$ . . . . .	176
VI.13	Modèle éléments finis de la plaque à 8 trous. . . . .	177
VI.14	Champ $E$ pour différents points de conception . . . . .	178
VI.15	Convergence de l'apprentissage - La plaque complète . . . . .	179
VI.16	Modes du champ pour la plaque complète . . . . .	180
VI.17	Iso-valeurs du champ pour la plaque complète . . . . .	180
VI.18	Indices de Sobol' - la plaque complète . . . . .	181
VI.19	Sensibilités FORM - Le cas de la plaque trouée . . . . .	182
VI.20	Schéma de la cuve . . . . .	183
VI.21	Champ thermique dans la cuve . . . . .	184
VI.22	Schéma du modèle de rupture . . . . .	185
VI.23	Résultats du calcul fiabiliste - Projet PROSIR . . . . .	186
VI.24	Résultats du calcul fiabiliste - Projet PROSIR . . . . .	188



# Liste des tableaux

III.1	Résultats des simulations et analyse FORM sur l'état-limite. . . . .	72
III.2	Résultats du calcul de fiabilité sur le modèle du fil conducteur. . . . .	94
III.3	Résultats du calcul de fiabilité sur le modèle du fil conducteur. . . . .	94
III.4	Résultats du calcul de fiabilité sur le modèle du fil conducteur. . . . .	109
IV.1	Définition des variables - Cas $R - S$ . . . . .	118
IV.2	Définition des variables - Surface bruitée . . . . .	119
IV.3	Définition des variables - la sphéroïde . . . . .	119
IV.4	Définition des variables - l'hyperbole . . . . .	120
IV.5	Résultats du calcul fiabiliste - Cas $R - S$ . . . . .	125
IV.6	Résultats du calcul fiabiliste - Surface bruitée . . . . .	126
IV.7	Résultats du calcul fiabiliste - la sphéroïde . . . . .	127
IV.8	Résultats du calcul fiabiliste - fonction convexe . . . . .	128
IV.9	Résultats du calcul fiabiliste - fonction concave . . . . .	128
IV.10	Résultats du calcul fiabiliste - hyperbole . . . . .	130
IV.11	Résultats du calcul fiabiliste - cas système . . . . .	130
IV.12	Résultats du calcul fiabiliste - cas système en 2D . . . . .	132
IV.13	Influence de la dimension - Cas Rackwitz . . . . .	133
V.1	Définition des variables - Poutre en flexion . . . . .	149
V.2	Définition des variables - Le treillis . . . . .	152
VI.1	Définition des variables - la sphère sous pression . . . . .	165
VI.2	Résultats du calcul fiabiliste - la sphère sous pression . . . . .	165
VI.3	Résultats du calcul fiabiliste - le treillis . . . . .	168
VI.4	Définition des variables - L'oscillateur non-linéaire . . . . .	170
VI.5	Résultats du calcul fiabiliste - oscillateur . . . . .	171



VI.6	Résultats du calcul fiabiliste - oscillateur . . . . .	173
VI.7	Résultats du calcul fiabiliste - la plaque trouée . . . . .	177
VI.8	Résultats du calcul fiabiliste - la plaque complète . . . . .	178
VI.9	Définition des variables - Projet PROSIR . . . . .	185
VI.10	Résultats du calcul fiabiliste - Projet PROSIR . . . . .	186
VI.11	Résultats du calcul fiabiliste - Projet PROSIR avec défaut aléatoire .	187

# Annexes

## Sommaire

---

Annexe 1. Les méthodes à noyaux . . . . .	209
Annexe 2. SQP . . . . .	211
Annexe 3. SVM et internet . . . . .	215

---



## Annexe 1. Les méthodes à noyaux ou Kernel trick

Petite annexe sur les méthodes à noyaux, très utilisées en statistiques... Le *Kernel trick*, traduit de Wikipedia [Wik06].

Utilisé en théorie de l'apprentissage, le *Kernel Trick* est une méthode permettant aisément de convertir un séparateur non linéaire en un séparateur linéaire, en projetant les observations ou points d'apprentissage de l'espace original vers un espace de dimension supérieure non linéaire (*feature space*) tel que la classification linéaire dans ce nouvel espace soit équivalente à une classification non linéaire dans l'espace original.

Ceci est rendu possible par l'utilisation du théorème de Mercer : toute matrice noyau semi-définie positive  $K(x, y)$  peut être exprimée en un produit scalaire dans un espace de grande dimension.

Plus spécifiquement, si les arguments du noyau  $K$  sont dans un espace mesurable  $X$  et que le noyau  $K$  est semi-défini positif alors, c'est-à-dire :

$$\sum_{i,j} K(x_i, x_j) c_i c_j \geq 0 \quad (\text{A1.1})$$

alors, quelque soit l'ensemble fini  $x_1, \dots, x_n$  de  $X$  et un ensemble fini de réels  $c_1, \dots, c_n$ , il existe une fonction  $\varphi(x)$  dont l'espace d'arrivée est un espace pré-Hilbertien (espace vectoriel muni d'un produit scalaire) de très grande dimension, tel que :

$$K(x, y) = \varphi(x) \cdot \varphi(y) \quad (\text{A1.2})$$

Le *kernel trick* transforme tout algorithme dépendant d'un produit scalaire entre 2 vecteurs. Si un produit scalaire est utilisé, il est remplacé par la fonction noyau. Cet algorithme non linéaire est équivalent à un algorithme linéaire opérant dans l'espace d'arrivée de  $\varphi(x)$ . Cependant, comme la fonction noyau  $K$  est utilisée, la fonction  $\varphi$  n'est jamais explicitée. Ce qui est préférable, puisque l'espace de dimension supérieure peut-être un espace de dimension infinie (comme dans le cas où la fonction noyau est gaussienne).

Le *kernel trick* a été publié historiquement par Aizerman *et al.* [AR64].

Cette technique est appliquée à différents types d'algorithmes en statistique et apprentissage :

- réseau de neurones : perceptrons ;
- Support Vector Machines ;
- analyse en composantes principales ;
- clustering.

La personne qui introduit le terme *kernel trick* est inconnue. Rappelons que le mot ”trick” signifie ruse et que la méthode constitue bien une ruse fondée sur la définition d’un noyau pour contourner la difficulté de la classification non linéaire.

## Annexe 2. SQP : Sequential Quadratic Programming

Cette annexe est destinée à la présentation de la résolution d'un problème d'optimisation quadratique linéaire. Ce type d'optimisation correspond à celle mise en place pour l'obtention des paramètres d'un SVM. L'optimisation la plus répandue consiste à utiliser le gradient pour se laisser conduire au point optimal recherché. Pour aller plus loin, la méthode quadratique utilise l'information du second ordre, en plus des gradients, pour améliorer la convergence. Cette méthode est donc très adaptée à l'optimisation de fonction quadratique, ce qui est le cas pour les SVM.

### Un problème d'optimisation quadratique

#### La forme générale

Un problème d'optimisation quadratique est une optimisation dont la fonction objectif est quadratique (par rapport aux variables d'optimisation...) et les contraintes sont linéaires, soit :

$$\min_x \frac{1}{2} x^T H x + g x \quad (\text{A2.3})$$

sous les contraintes :

$$\begin{cases} Ax - b = 0 \\ Cx - d \leq 0 \end{cases} \quad (\text{A2.4})$$

où  $H$  est une matrice symétrique constante,  $A$  et  $C$  sont des matrices constantes et  $g$ ,  $b$  et  $d$  sont des vecteurs constants.

#### Résolution du cas sans contrainte

La solution d'un problème QP sans contrainte avec  $H$  définie positive est évidente :

$$x^* = -H^{-1}g \quad (\text{A2.5})$$

### Résolution du cas avec contraintes d'égalité

La solution du problème dans le cas de contraintes d'égalité du problème QP où  $H$  est définie positive peut être obtenue en regardant le lagrangien associé :

$$L = \frac{1}{2}x^T Hx + g^T x + \lambda^T (Ax - b) \quad (\text{A2.6})$$

Le gradient de ce Lagrangien s'écrit :

$$\nabla L = Hx + g + A^T \lambda = 0 \quad (\text{A2.7})$$

En associant cette dernière équation avec la contrainte égalité  $Ax - b = 0$ , on obtient un système d'équations linéaires :

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -g \\ b \end{bmatrix} \quad (\text{A2.8})$$

La solution est alors :

$$\begin{cases} x^* = -H^{-1}(g + A^T \lambda) \\ \lambda^* = -(AH^{-1}A^T)^{-1}(b + AH^{-1}g) \end{cases} \quad (\text{A2.9})$$

### Résolution du cas avec contraintes d'inégalité

Si le problème QP contient des contraintes d'inégalités, alors l'algorithme à mettre en place va intégrer un système évolutif dans lequel une série de problèmes avec des contraintes d'égalité vont être résolues.

$$\min_d \frac{1}{2}(x + d)^T H(x + d) + g(x + d) \quad (\text{A2.10})$$

sous les contraintes :

$$a_i(x + d) - b_i^* = 0, i \in W_k \quad (\text{A2.11})$$

où  $d$  est la direction de recherche.  $W_k$  contient toutes les contraintes égalités et les contraintes d'inégalités actives pour le pas d'itération en cours. Cet ensemble de contraintes est mis à jour à chaque itération selon un critère défini.

### La méthode SQP

#### L'écriture générale d'un problème d'optimisation

Nous revenons à un cas plus général d'optimisation pour présenter une variante de la méthode SQP. La plupart des problèmes d'optimisation sous contraintes non

linéaires peuvent s'écrire sous la forme suivante :

$$\min_x f(x) \quad (\text{A2.12})$$

sous les contraintes :

$$\begin{cases} h(x) = 0 \\ g(x) \leq 0 \end{cases} \quad (\text{A2.13})$$

où :

$$g : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ et } h : \mathbb{R}^n \rightarrow \mathbb{R}^p \quad (\text{A2.14})$$

### L'algorithme SQP

L'algorithme SQP est l'un des plus répandu pour résoudre un problème d'optimisation sous contraintes non linéaires. Les conditions d'optimalité de Karush Kuhn Tucker sont appliquées de manière itérative. Un sous-problème d'optimisation quadratique est résolu à chaque itération majeure, et la solution donne la direction de recherche. Une optimisation linéaire est réalisé dans la direction de recherche. À chaque itération majeure, le Hessien est mis à jour par la méthode BFGS.

Le problème QP résolu à chaque itération est une approximation du problème original où l'objectif est quadratique et les contraintes linéaires :

$$\min_d f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 L(x_k) d \quad (\text{A2.15})$$

sous les contraintes :

$$\begin{cases} g_i(x_k) + \nabla g_i(x_k)^T d \leq 0 \\ h_i(x_k) + \nabla h_i(x_k)^T d = 0 \end{cases} \quad (\text{A2.16})$$

où le Lagrangien  $L$  est défini par l'expression :

$$L(x) = f(x) + \lambda^T g(x) + \mu^T h(x) \quad (\text{A2.17})$$

$\lambda$  étant le vecteur des coefficients de Lagrange pour l'optimisation et  $\mu$  le vecteur des coefficients des conditions de KKT.

À la place de l'utilisation du Hessien du Lagrangien (trop coûteuse), une approximation est utilisée :  $\nabla^2 L(x)$  est remplacé par  $H_k$  qui est l'approximation BFGS du Hessien.

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T p_k} - \frac{H_k p_k p_k^T H_k}{p_k^T H_k p_k} \quad (\text{A2.18})$$

où :

$$\begin{cases} p_k = x_{k+1} - x_k \\ q_k = \nabla f(x_{k+1}) + \lambda^T \nabla h(x_{k+1}) - (\nabla f(x_k) + \lambda^T \nabla h(x_k)) \end{cases} \quad (\text{A2.19})$$



La solution du sous problème QP permet d'obtenir une direction de recherche  $d_k$  et ainsi obtenir le point de l'itération suivante :

$$x_{k+1} = x_k + \alpha d_k \quad (\text{A2.20})$$

Il s'agit ensuite d'optimiser la valeur de  $\alpha$  pour améliorer la convergence.

### Les étapes de l'algorithme SQP

1. À partir du point  $x_k$  et de l'approximation du Hessien en cours  $H_k$ , le sous-problème QP est résolu pour obtenir une direction de recherche  $d_k$ . La solution du sous-problème nous donne une estimation des multiplicateurs  $\lambda$  et  $\mu$ .
2. À partir de la direction  $d_k$ , le problème d'optimisation de la direction est résolu. Il permet l'obtention du point  $x_{k+1}$ .
3. L'approximation du Hessien  $H_{k+1}$  est mise à jour à partir de la formule BFGS.
4. Vérification du critère de convergence et retour à l'étape 1 en cas de non convergence.

## Annexe 3. L'apprentissage statistique sur le web

Cette annexe est destinée à répertorier les sites internet contenant énormément d'informations et parfois des publications sur l'apprentissage statistique et en particulier sur les SVM.

Vous y trouverez :

- des informations théoriques ;
- des tutoriaux ;
- de l'information sur les logiciels ;
- des applications libres ;
- ...

### SVM

<http://www.isis.ecs.soton.ac.uk/resources/svminfo/>

<http://www.kernel-machines.org/>

<http://www.kernel-machines.org/>

<http://www.learning-with-kernels.org/>

<http://svms.org/>

<http://www.cs.cmu.edu/~schneide/tut5/node1.html>

<http://www.data-compression.com/vq.shtml>

[http://www.support-vector-machines.org/SVM\\_soft.html](http://www.support-vector-machines.org/SVM_soft.html)

<http://chopin.zet.agh.edu.pl/~wichtel/>

<http://www.kyb.tuebingen.mpg.de/bs/people/spider/>

<http://www.ai.mit.edu/courses/6.867-f04/lectures/>

<http://www.cs.wisc.edu/dmi/lsvm/>

<http://www.boosting.org/>

<http://www.data-compression.com/vq.shtml>

<http://asi.insa-rouen.fr/~arakotom/toolbox/index.html>

[http://www.telos-eu.com/qui\\_sommes\\_nous.php](http://www.telos-eu.com/qui_sommes_nous.php)

<http://obousquet.googlepages.com/mldmcompanies>

<http://obousquet.googlepages.com/>

<http://eprints.pascal-network.org/view/subjects/LO.html>

<http://www.pertinence.com/about/pl.html>

[http://eric.univ-lyon2.fr/ricco/cours/supports\\_data\\_mining.html](http://eric.univ-lyon2.fr/ricco/cours/supports_data_mining.html)

### **fiabilité**

FERUM [www.ce.berkeley.edu/FERUM](http://www.ce.berkeley.edu/FERUM)

<http://bwk.kuleuven.be/> site belge... cas de validation...





## Résumé

La prise en compte de l'incertain dans les analyses mécaniques est une condition indispensable pour un dimensionnement optimal et robuste des structures. Avec cet objectif les méthodes de couplage mécano-fiabiliste doivent intégrer les modélisations mécaniques de plus en plus complexes (comportement non-linéaire, dynamique, fatigue, mécanique de la rupture, etc), complexité qui rend les analyses de plus en plus exigeantes en temps de calcul.

La voie retenue dans la thèse consiste à substituer au modèle mécanique, un modèle approché obtenu par apprentissage statistique. La mise en œuvre s'appuie sur la méthode des *Support Vector Machines* (SVM). Deux nouvelles méthodes sont proposées. *SMART Support vector Margin Algorithm for Reliability esTimation* permet d'élaborer de manière incrémentale une base de données optimale pour l'apprentissage de la frontière entre sûreté et défaillance. En couplant l'apprentissage avec la technique des *subsets simulations*, <sup>2</sup>*SMART Subset by Support vector Margin Algorithm for Reliability esTimation* prend en compte de faibles niveaux de probabilité et un nombre relativement important de variables.

Les travaux ont été validés sur des cas tests issus de la littérature ainsi que sur des cas mécaniques plus significatifs issus de problèmes industriels. Les résultats obtenus confirment la robustesse des méthodes proposées pour traiter les analyses de fiabilité et de sensibilité.

## Abstract

Uncertainties usually play an important role in structural engineering and their effects need to be evaluated for robustness and safety. Coupling mechanical and stochastic data models for complex systems using a classical procedure may become prohibitive in terms of computer time. This time increases as the probability of failure of interest decreases, and this is a major limitation in using these methods for industrial applications. To overcome the computing time difficulty, limit-state substitution is proposed.

J.E. Hurtado (Structural Reliability - Statistical Learning Perspectives) introduces Support Vector Machine to reliability and applies their classification properties to produce an efficient Monte-Carlo simulation. In SMART, we improved his algorithm to build the limit-state surrogate for reliability application. By merging Support Vector Machine classification into subset simulation, we propose here a robust method for reliability analysis referred as <sup>2</sup>SMART for Subset by Support vector Margin Algorithm for Reliability esTimation.

The proposed methodology is detailed and its efficiency is illustrated through various examples, from test cases for validation to structural mechanics applications.